Firewalls

and

Virtual Private Networks

A thesis

submitted in partial fulfilment

of the requirements for the Degree

of

Master of Commerce in Computer Science

in the

University of Canterbury

by

B. A. Harris

University of Canterbury

1998

To Sharalie,

with love.

Acknowledgements

There are a great number of people that I wish to thank for helping me undertake this thesis. Perhaps I should start with my supervisor and friend, Ray Hunt, who has provided me with unwavering support, encouragement, and of course much needed criticism. I believe that without your help and enthusiasm this work would probably have never seen the light of day. Thank you.

I am equally indebted to my fiancee Sharalie, who has stoically endured many months of separation while I pursued my dream. Your love and encouragement were given without hesitation, and for that I love you even more.

I cannot go any further without acknowledging all of the support and encouragement given by my parents, sister, and future parents-in-law. Each of you have contributed in your own special way, and helped smooth a path that at times threatened to become very difficult indeed.

There are a number of friends that I wish to thank because they have contributed significantly, in a variety of ways, to the completion of my thesis. So please bear with me — Matthew Brady for commanding and conquering my battles; Greg Slui and Rebecca Gilmore for being great friends and providing me with somewhere to go; Mark Snelling for inspiration and cryptographic support; Stephen Harvey for entertaining discussion and helping me understand public-key infrastructures; and Sylvester and Scooby-Doo for providing essential relaxation and escapism.

Finally, a very special mention is deserved for Malcolm Shore of the Government Communications Security Bureau. Malcolm has provided a great deal of technical direction and support, and was instrumental in arranging for me to take leave from work to return to university and complete my thesis. Thank you.

Funding for this thesis was provided by the Government Communications Security Bureau.

Brendon Harris
1998

Table of Contents

List of Figures

List of Tables

Abstract

The Internet has become a global computing phenomenon, and during the 1990's has had more influence on the computer – communications industry than any other development in its history. There are two major issues effecting the development of the Internet for the 21st century; performance and security. This thesis is concerned with the later; in particular the issues raised by the interconnection of TCP/IP based networks between trusted and untrusted network domains.

Four main topics are addressed: the common threats and vulnerabilities that effect the TCP/IP protocol suite at the Network, Transport, and Application layers; the application of firewall architectures to counter the risks posed by TCP/IP based connections between trusted and untrusted network domains; the issue of independent firewall architecture evaluation and certification; and the application of Virtual Private Network (VPN) technology to protect traffic over untrusted networks.

This thesis examines the common threats and vulnerabilities which effect the current TCP/IP protocol suite, and hence the Internet. A firewall architecture can be a powerful tool for preventing attacks based on TCP/IP vulnerabilities, however, it is only as effective as the security policy that it implements. Although firewalls can benefit computer and network security, they suffer from several significant limitations, including; the inability to protect network traffic; defending against insider abuse; and controlling the content of end-user access (e.g. virus infected files, Java applets, etc.)

Firewalls are generally considered impregnable, however they are certainly not immune to software and hardware vulnerabilities. Therefore, this thesis examines independent evaluation and certification of firewall architectures with particular focus on New Zealand and Australian efforts.

The final section of this thesis examines the use of VPNs for securing network traffic. The amalgamation of VPN and firewall technologies allows the security policy to be extended onto the network in the form of services, such as, confidentiality, integrity, non-repudiation, and strong authentication.

Chapter 1. Internet Beginnings

1.1 Introduction

Today the term "Internet" suggests an image of a global network of computers that exists solely to market products and services, and generate profit for the organisations they represent. Fortunately, the Internet is far more interesting and varied than the image presented by increasing commercialisation. The Internet has grown from its small beginnings in the late 1960's into a truly global network, not just in terms of geography but also in respect to national boundaries.

The purpose of this Chapter is to develop an image of the Internet in its early days before commercialisation, and introduce some important concepts which provide background, and help in the understanding of many of the topics throughout this thesis.

1.2 The ARPANET

The Internet has evolved into its present form from the work begun by the US *Defense Advanced Research Projects Agency* (DARPA) in the late 1960s. In 1969, DARPA sponsored a project that became known as ARPANET, whose rationale was to provide high-bandwidth connectivity between major government, education, and research computing establishments [Hare et al., 1996]. ARPANET was an experiment in *packet-switched*[1] network communications, aimed at providing the US *Department of Defense* (DoD) with a command and control network capable of delivering data to its destination even if some of the intervening network segments were disabled through, for example, a nuclear attack.

Throughout the 1970's DARPA continued to fund ARPANET, which was extended to include experimental satellite and radio communication links. From this came the development of a common framework of networking technologies, out of which emerged the *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP), and the *Internet Protocol* (IP). These three protocols provide the foundation for most of the applications which make use of the Internet today.

The key protocol is IP, which provides a common address space that allows messages, known as datagrams[2], to be delivered between the many separate networks that constitute the Internet. TCP allows extremely reliable data transmission over possibly unreliable networks. UDP, which is analogous to postal mail, transmits discrete collections of data without any guarantee of delivery. Both TCP and UDP require IP, and so do all other protocols that are part of the TCP/IP protocol suite. Note that the acronyms TCP/IP and IP are often used to refer to the whole protocol suite.

It is important to mention here that the development of TCP/IP cannot be attributed entirely to the US research the community. In fact, Norway and England were involved from the beginning in the development of IP, while France and the United Kingdom (UK) provided considerable technical contributions to TCP and IP. For example, the original TCP retransmission algorithm was known as the RSRE (Royal Signals and Radar Establishment) algorithm in name of the UK organisation that developed it [Carl-Mitchell et al., 1993].

The initial protocols used on ARPANET were not TCP/IP, instead TCP/IP was invented using ARPANET. The basic protocol used on ARPANET was called *Network Control Protocol* (NCP) and the protocol for a host to communicate with a router was called *BBN 1822*, after the technical report

[1] Packet-switching involves breaking data into discrete units for transfer over a network, each packet is routed from one computer to the next until the destination is reached. In most cases a dedicated computer, known as a *router*, is used to relay packets between the network segments that it connects.

[2] A datagram is a discrete chunk of data that has sufficient addressing information for it to be routed independently in an internetwork. Note: the term "packet" is synonymous with the term "datagram".

that specified it. Aside from research into packet-switching, the intended use for ARPANET was to provide users with the ability to remotely login to very expensive computer resources, and save ARPA from duplicating them at each research facility. To enable this, the *Virtual Terminal Protocol* (Telnet) and the *File Transfer Protocol* (FTP) were developed. The NCP versions of these protocols were modified for use with TCP/IP, and have proved so successful that they are still among the most popular Internet protocols today.

By January 1983, all computers connected to ARPANET were using the TCP/IP protocols. In fact, ARPANET had become so successful that it was no longer considered an experimental network and operational control was passed over to the *Defense Communications Agency* (DCA), now known as the *Defense Information Systems Agency* (DISA). A non-experimental internet, known as ARPA Internet, was begun in January 1983 by DCA who required all connected nodes to use TCP/IP. At the same time, DCE divided ARPANET into two networks; the ARPANET for research, and MILNET for military operations.

Today's popularity of the TCP/IP protocol suite can be traced back to its initial implementation in the University of California at Berkeley's 4.2BSD version of UNIX. The TCP/IP protocols were included because ARPA had partly funded the development of 4.2BSD to provide them with a research platform. As funding had come from public sources, including the State of California, 4.2BSD was made available for the cost of its distribution and therefore spread quickly. Ever since the development of UNIX and TCP/IP has been closely entwined.

Although ARPANET provided the first backbone network of the early Internet (until retirement in 1990) it was still predominantly a government network. Thus, in 1994 the National Science Foundation created a network backbone across the US known as NSFNET, which also used TCP/IP. NSF also provided initial funding for smaller regional networks, which are commonly referred to as "NSFNET regionals." These regional networks now provide extensive connectivity to universities, government agencies, and commercial businesses.

ARPANET, MILNET formed the earliest US nation-wide network backbones, while others were added by US government agencies, such as the *National Aeronautics and Space Agency* (NASA). In addition to US based agencies and educational institutions, 1973 saw the first European organisations connect to ARPANET. Over the next decade a number of Japanese and European networks developed and were connected, including the European EUNET, and the UK Joint Academic Network (JANET) [Sim et al, 1997]. With the participation of other government agencies, the name given to this conglomeration of networks was changed, from ARPA Internet to Federal Research Internet to TCP/IP Internet and finally to just the Internet.

1.3 New Zealand and the Internet

New Zealand's history in relation to the Internet begins in 1985 with the establishment of a 2400-bit.sec^{-1} dial-up connection from the University of Victoria in Wellington to the University of Calgary in Canada. This link was used to transfer electronic-mail (email) using UUCP (UNIX-to-UNIX Copy), a store and forward protocol for linking together UNIX systems over low-speed serial lines. Due to the low-speed and expense of such a connection, other resources such as USENET news archives continued to be delivered on magnetic storage tapes by airmail [Wiggin, 1996].

A similar link to the University of Waterloo in Canada was established by the University of Canterbury. At this time no direct links existed between Universities within New Zealand, instead email between Victoria and Canterbury universities was delivered via Canada! However, in 1986 these links were replaced with packet-switched links to the University of Melbourne in Australia and a US site. Sufficient capacity was now available to carry both email and USENET news feeds to the Universities and other interested organisations within New Zealand.

The University of Waikato established the first true IP-based Internet connection from New Zealand to the US in April 1989, with a 9600 bit.sec^{-1} analogue link to Hawaii. This was as a joint development

project with NASA and formed part of the Pacific Communications program (PACCOM) [Neal, 1996]. The gateway itself was known as "the PACCOM gateway," but was soon renamed "NZGate."

It is interesting to note that the New Zealand government at that time provided no subsidy whatsoever to establish the local end of the Waikato link. This was in stark contrast with NASA which provided a generous subsidy to establish and operate the US end in Hawaii. Instead, funding was provided by six of the New Zealand universities, with each agreeing to provide one-sixth of the start-up and on-going costs which where in-turn passed on to the users. Regardless of this "user-pays" approach, the first four years saw significant growth with the link speeds increasing from 9600 bit.sec^{-1} to 512 Kbit.sec^{-1}.

In addition to the Waikato Internet connection, the period from 1990 to 1992 saw the beginning of IP links between the New Zealand Universities. The Maori word "Kawaihiko"[3] was chosen to name the network connecting all seven universities (i.e. Canterbury, Otago, Lincoln, Victoria, Massey, Waikato, and Auckland).

The Kawaihiko network began operating in April 1990, and from the beginning greatly improved communications between the universities. Initially each site had a Cisco router, and was connected to the next node by a 9600 bit.sec^{-1} *Digital Data Service* (DDS) link. The original topology consisted of a central triangle joining Waikato, Victoria and Canterbury, with the others connected to these (see Figure 1-1). Thus, the first large scale IP network had arrived in New Zealand, and began introducing many users for the first time to the benefits of TCP/IP services, such as email, FTP, Telnet, and USENET news.

During the early 1990's government research institutions, such as the *Department of Scientific and Industrial Research* (DSIR), were also installing IP networks to connect their main sites. By April 1991 the DSIR had connected five of its sites, i.e. Gracefield, Wellington, Palmerston North, Christchurch and Hamilton, which was known as DSIRnet. At the same time the *Ministry of Agriculture and Fisheries* (MAF) had installed their own IP network (using Telecom's X.25 packet-switched network, or PACNET), known as MAFnet.

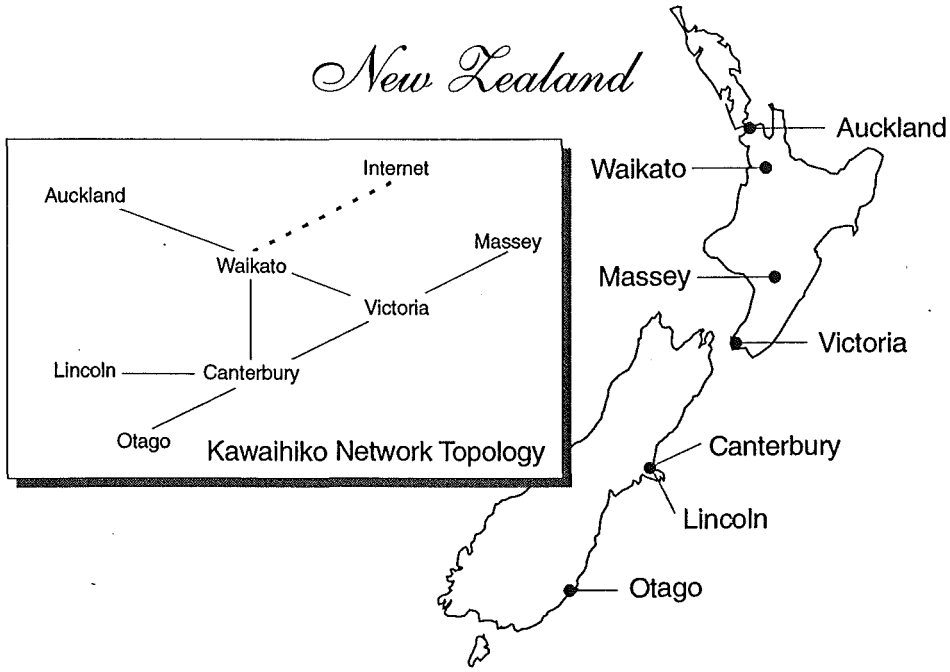

Figure 1-1 The location of each university in New Zealand, including the logical topology of the Kawaihoki network (see inset).

[3] "Kawaihiko" is a Maori word, derived from *kawai* (a branching structure, like tree roots) and *hiko* (electricity).

Prior to 1992 there were three Research and Education networks in New Zealand; DSIRnet, MAFnet, and Kawaihiko. However, in 1992 the government restructured the DSIR and MAF creating eleven partly independent *Crown Research Institutes* (CRIs), and a single co-ordinated network known as CRInet.

By May 1992 it had been decided to combine the existing Research and Education networks into a single network providing a national backbone based on Frame Relay, but retaining DDS links where Frame Relay was too expensive. Thus between May and July of 1992, CRInet, Kawaihiko, the National Library network, and the *Ministry of Research, Science and Technology* (MoRST) network were combined into the New Zealand *National Research and Educational Network* (NREN) [Brownlee, 1994].

Initially NREN was managed through an informal arrangement, which was eventually replaced by the Tuia[4] Society that resulted in NREN being renamed to TuiaNet. Tuia is an incorporated society, providing its members (e.g. the CRIs, Universities, etc.) with a legal entity and a formal structure within which to run the single national backbone network. Tuia retained the original network groupings as management groups, thus, Kawaihiko is the Tuia management group which co-ordinates inter-university networking, while Industrial Research Limited (IRL) and AgResearch co-ordinate groups of CRIs which correspond to the old DSIR and MAF networks. The remaining sites (e.g. National Library, and MoRST) have remained as single-member management groups.

Connections to the Internet have continued to grow dramatically since 1992, especially since the deregulation of the telecommunications market. Deregulation provided a competitive environment for independent *Internet Service Providers* (ISPs), and has helped bring Internet access to all parts of New Zealand. To help foster this competitive environment, and in response to the increasing pressures of maintaining an international link, early 1996 saw Waikato University turn management of NZGate over to Netway (a subsidiary of Telecom) and Clear Communications. NZGate is now known as the *New Zealand Internet Exchange* (NZIX). In addition, NZIX is now no longer the sole Internet gateway, a number of other organisations, such as IBM, Voyager, and CompuServe, operate their own international links.

1.4 Internet Growth

There are a number of factors contributing to the phenomenal growth of the Internet in New Zealand and around the globe. One is an increasing recognition and acceptance of the Internet in home based computing. Another is the growing commercialisation of the Internet to support business initiatives, such as advertising, electronic commerce, and organisations basic day to day functioning.

The reason for such growth during the 1990's can be attributed to the development, in 1989, of the *World Wide Web* (WWW) by Tim Burners-Lee of CERN. It was this single application that transformed the Internet into a global, multimedia information service, and helped attract a much broader spectrum of users. As of April 1998, a survey conducted by Nua Ltd.[5] estimated that the number of Internet users was in excess of 115 million — representing 2.4% of the world's population.

A 1998 survey conducted by Network Wizards[6], shown Figure 1-2, page 5, depicts the explosive growth of the Internet since 1991. The survey measures the number of hosts, i.e. connected computers, on the Internet. However, these numbers are generally considered to be an underestimate of the true size

---

[4] "Tuia" is a Maori word which means "bound together."

[5] Nua Ltd. Internet Population survey results are available at **http://www.nua.net/surveys/how_many_online/world.html**

[6] Network Wizards Internet Domain survey results are available from **http://www.nw.com**. It is important to note that the methodology used to conduct the January 1998 survey was changed in an effort to achieve more accurate results. The plot of "Adjusted Survey Results" is an attempt to adjust the survey results from January 1995 to July 1997 to reflect the new methodology and assume a previous history. However, no direct comparisons can be made between the old and new results.

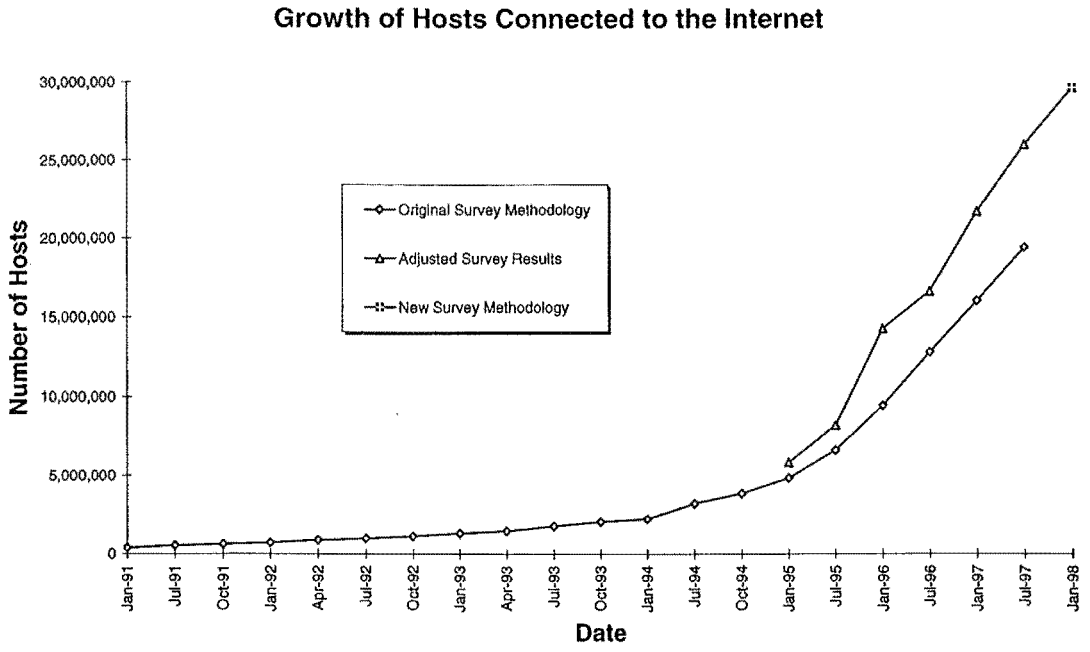

Figure 1-2 Growth of host machines on the Internet.

due to the fact that firewalls prevent the accurate surveying of a considerable number of network domains. From a New Zealand perspective the survey found that 169,264 hosts were connected under the second level *.nz* domain, which represents 100% growth since the January, 1997, survey!

Prior to 1990 the Internet was predominantly used for data exchange and communication between academics and government researchers. It is now used widely from the home and office to access the

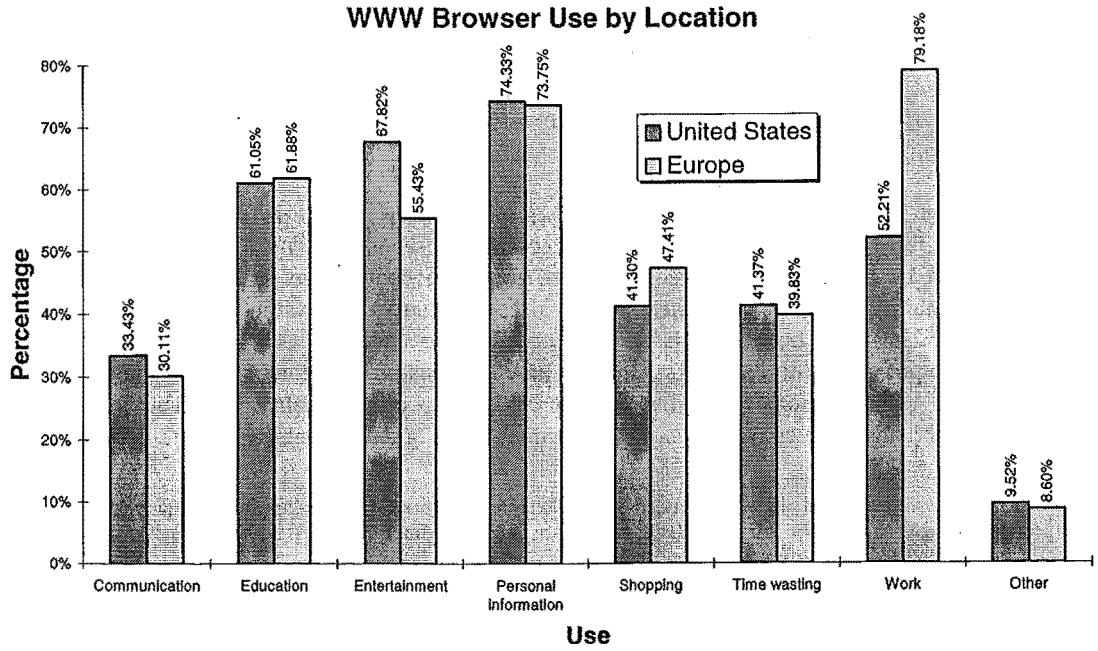

Figure 1-3 Comparison of WWW-browser use by US and European users.

vast repository of information published on the Internet. This expansion has been influenced by the increased power of home and office computers which has driven the development of on-line shopping, 3-dimensional virtual reality worlds, and real-time audio and video applications including Internet telephony and video-conferencing. Figure 1-3 is based on results from the 8th GVU (Graphics, Visualization & Usability)[7] Survey, and compares the WWW-browsing habits of users in the US and Europe. Another interesting finding from the GUV Survey is the types of Internet technology currently being used by users in the US and Europe (see Figure 1-4).

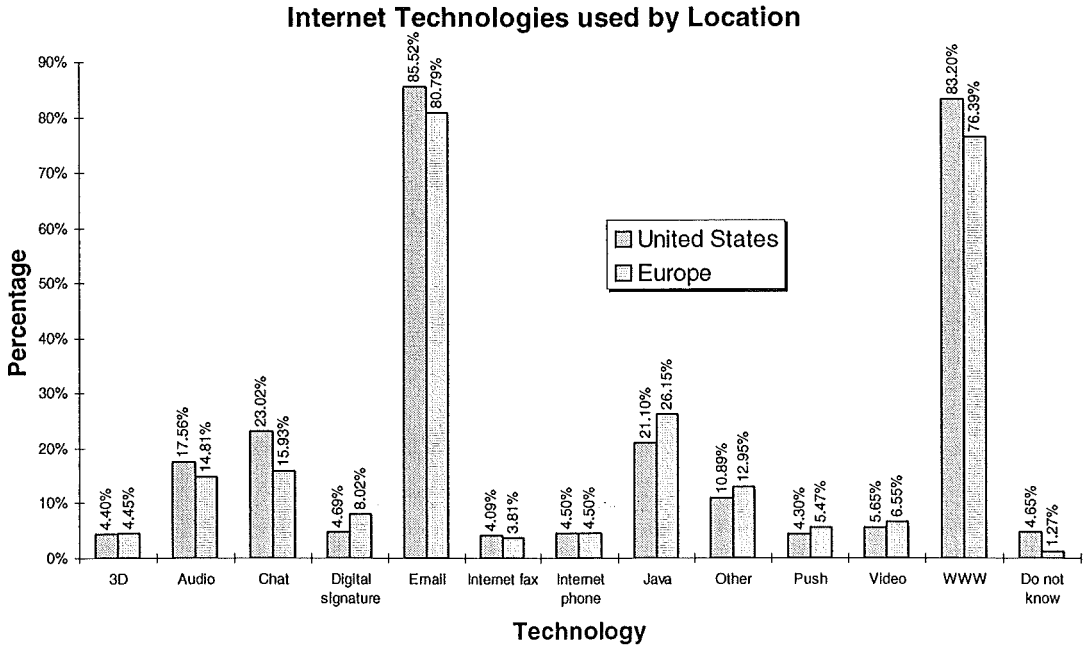

Figure 1-4 Comparison of Internet technologies used by US and European users.

Although Figure 1-4 categorises a wide range of Internet technologies, it is clear that WWW-browsing and Email are by far the most dominant. This trend is supported in Figure 1-5, page7, which presents results from the 1997 UK Internet Survey[8], which focuses specifically on the services (or applications) used to access information on the Internet.

The information resources contained on the Internet are immense, it is possible to search for information on the most obscure topic and yet receive thousands of "hits"[9]. The ease with which information can be found and retrieved is one of the main contributors to the Internet's growth. Obviously, the ability to search and retrieve results almost instantly is a great advantage for researchers and academics. However, businesses have also found it very valuable as a marketing tool, which enables them to provide up-to-date information to new and existing customers without the delays and costs associated with traditional paper based resources such as product catalogues. The ability of the WWW to support dynamic content is especially valuable for organisations (e.g. share-brokers, travel

[7] GVU surveys are conducted by the Georgia Tech Research Corporation, the results are provided online at **http://www.gvu.gatech.edu/user_surveys**. The 8th survey was conducted from October 10, 1997 through to November 16, 1997, and represents responses from 7200 respondents.

[8] The UK Internet Survey was conducted by the Red Square Group Ltd., the results were released on December 1, 1997, and represents the response of 768 respondents. The results are freely available at **http://www.redsquare.co.uk/index.htm**

[9] A "hit" is Internet jargon used to describe the successful outcome of a search. Information repositories, such as WWW-pages, are continually indexed and categorised by powerful search engines which can then be queried with a set of key words entered by the user. Results are usually returned to the user in the form of hyper-links on a dynamically generated WWW-page.

agents, real-estate agents, etc.) that deal with continually changing, or real-time, information that must be constantly updated for customers.

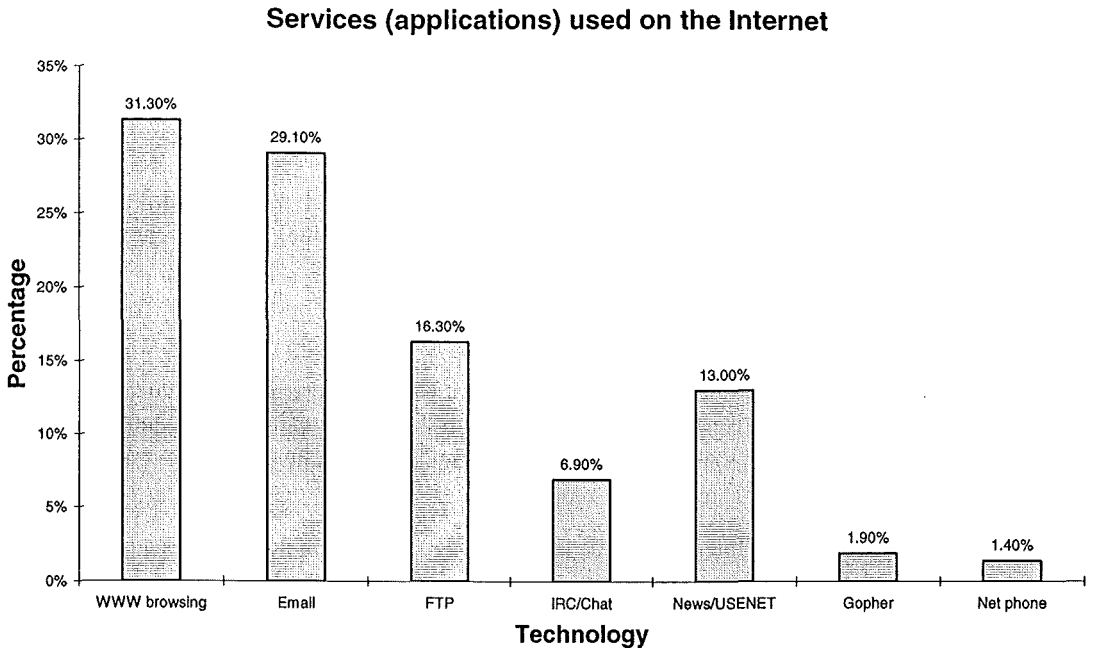

Figure 1-5 Services used by Internet users in the UK.

In addition to providing information, many businesses are using the Internet to provide better customer service. For example, most software and hardware manufacturers maintain WWW-pages which allow their customers to download the latest patches, service releases, or device drivers. Selling and distributing intangible products (e.g. software, electronic books and reports, etc.) on the Internet can help manufacturers/suppliers reduce the cost of distributing their products. For example, instead of buying a software product from the local computer store, a customer can go directly to the supplier's WWW-site and download it. The customer then, pays for the majority of the distribution costs (i.e. download and storage), and the manufacturer no longer has to ship the product to re-sellers, nor provide physical packaging and distributable media (e.g. floppy-disk, compact-disk).

The Internet has in essence become the next business frontier, reaching every continent and permeating all manner of organisations. However, the use of the Internet for commerce, such as banking and shopping, is hindered by the lack of security mechanisms to protect the transactions. Many people do not use their credit-card on the Internet because of their fear of becoming a victim of fraud. To placate such concerns Visa and MasterCard are jointly developing the *Secure Electronic Transaction*[10] (SET) protocol to secure credit-card transactions over open networks (e.g. the Internet). On December 19, 1997 a new corporate entity called SET Secure Electronic Transaction LLC (or SETCo), was formed by Visa and MasterCard to provide the structure that will govern and direct the future development of the SET protocol, as well as other key functions that are required to support its implementation. However, in the interim many Internet businesses are securing customer transactions with the Secure Socket Layer (SSL) protocol developed by Netscape Communications (see Chapter 7 for a discussion of SSL). SSL provides a private connection between the customer and the supplier, over which sensitive transactions such as customer details and credit-card information can be sent.

Due to the Internet's growth and increasing commercialisation, the Internet now consists of many interconnected networks which belong to a myriad of public, private, and government organisations.

[10] Information on SET and its implementation is available at **http://www.visa.com/cgi-bin/vee/nt/ecomm/set/main.html**

Unfortunately not all are friendly, some may harbor attackers while some may practice traffic monitoring. It is not possible to send information over a "safe" path because of the very reasons which make the Internet so resilient — paths change constantly to deal with network outages or other problems, so even after a connection is established between two points the information transmitted may not follow the same path. Finally, a problem which stems directly from the Internet's explosive growth, is that the number of attackers is potentially growing at an equivalent rate!

1.5 Summary

The Internet has grown from connecting a mere 200 or so hosts in 1981, to nearly 30 million at the start of 1998. New Zealand has also seen startling growth and had nearly 170,000 Internet connections at the beginning of 1998. Since 1981 the Internet has changed from a network serving only academic and research interests, to a truly global network supporting the interests of individuals to multinational organisations.

In fact, increasing numbers of businesses and institutions are using the Internet to conduct their day-to-day business. It is becoming increasingly difficult for organisations to remain isolated and yet continue to do business with their Internet connected partners. As the Internet continues to grow, so will the role it plays in peoples everyday lives.

Unfortunately, the growth of the Internet and the increasing dependence on it, have presented many new threats and security challenges. The remainder of this thesis deals with specific technologies that can be used to address these security threats — in particular *firewalls* and *virtual private networks* (VPNs).

Chapter 2. Overview of TCP/IP

2.1 Introduction

As previously mentioned the term TCP/IP is used to refer to the collection of communications protocols that have evolved out of the initial DARPA project. The 1990's have seen the TCP/IP suite become the most widely accepted networking architecture.

Although many protocols contribute to the TCP/IP suite this Chapter only discusses those whose understanding is necessary for the remainder of this thesis. For a detailed discussion of these and other protocols in the TCP/IP suite the reader should consult [Carl-Mitchell et al., 1993] [Stevens, 1994] and the relevant *Requests for Comments*[11] (RFCs).

2.2 The TCP/IP Protocol Suite

The TCP/IP suite recognises that communication between heterogeneous computer systems is a complex and diverse problem. One which cannot be accomplished by a single all-encompassing protocol [Stallings, 1991]. Therefore the task of communicating is divided between a series of modules, which are layered on top of each other in a hierarchical manner (see Figure 2-1).

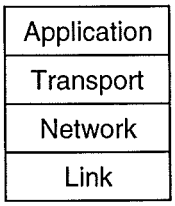

Figure 2-1 The four layers of the TCP/IP protocol stack.

Each layer provides a different functional requirement [Stallings, 1991] [Stevens, 1994]:

- *Link-Layer* – contains the protocols which provide access to a communication network. This usually consists of the host operating system's device driver, and the *network interface card* (NIC). The Link-layer handles all aspects of physical interfacing with the communications medium. Protocols at this level route data between hosts attached to the same network segment, and may provide additional services, such as, flow and error control.

- *Network-Layer* – provides the functionality which allows data to traverse separate logical networks between hosts. Therefore, the Network-layer provides the internetwork routing function. At this layer there are three different protocols; IP, *Internet Control Message Protocol* (ICMP), and the *Internet Group Message Protocol* (IGMP). Protocols at this layer can be implemented in either host or network devices (e.g. routers).

[11] The Requests for Comments (RFCs) are a series of notes, started in 1969, about the Internet (originally the ARPANET). The notes discuss many aspects of computing and computer communication focusing on networking protocols, procedures, programs, and concepts, but also including meeting notes, opinion, and sometimes humour. However, RFCs are also used to publish specification documents relating to the Internet protocol suite, as defined by the *Internet Engineering Task Force* (IETF) and its steering group (the IESG). The official RFC Editor WWW-page which provides links for searching repositories of RFCs is available at **http://www.isi.edu/rfc-editor/**

- *Transport-Layer* – provides functionality which enables data to be delivered between two applications on different host computers. There are two important protocols at the Transport-layer; the TCP, and the UDP.

- *Application-Layer* – handles the details of the applications.

The relationships between the common protocols at each layer of the TCP/IP suite are shown in Figure 2-2 (adapted from [Stevens, 1994]). Unfortunately, the term "TCP/IP" promotes a misconception that TCP is solely dependant on IP, and further suggests that all applications rely on TCP. In reality, TCP can exist quite happily over any other internetworking protocol and was in fact designed with this objective in mind [Postel, 1981d]. Figure 2-2 clearly dispels any illusion that applications depend solely on TCP, in fact, many applications are more efficient using connectionless UDP than they would be using TCP, while some would not even be possible. The term "TCP/IP" is used throughout this thesis to infer the dependence of TCP on IP, and to refer to the TCP/IP suite in general. (Note, in most cases the surrounding context will clarify the meaning, however, where ambiguity could have arisen additional clarification was included.)

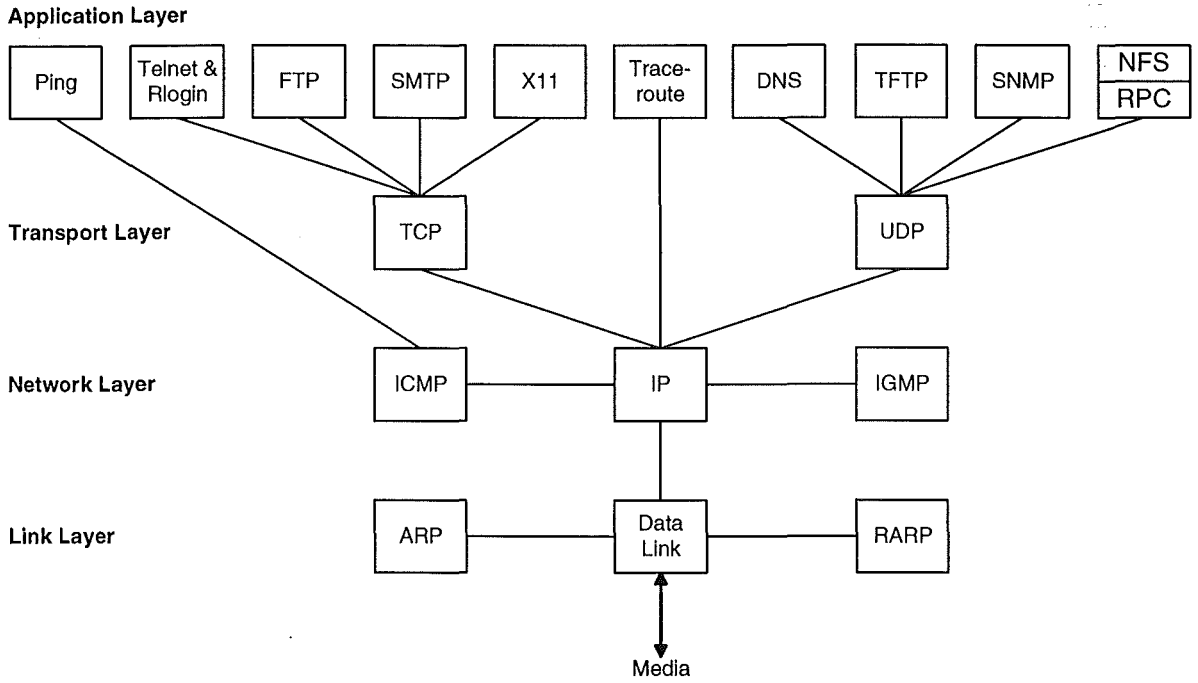

Figure 2-2 Relationship of protocols in the TCP/IP protocol suite.

Although TCP and IP are the dominant internetworking protocols on the Internet, other protocols have and are being developed. One in particular, the *Open Systems Interconnection* (OSI) reference model, was adopted in 1983 by the *International Organisation for Standardisation* (ISO). Essentially it provides a framework for defining standards for connecting heterogeneous computers, and like the TCP/IP suite[12] the OSI protocol stack is layered. However, the OSI layering is far more abstract and consists of seven layers. This has proved, in practice, to be restrictive and complicated to implement. Therefore, the majority of interconnected networks have continued to implement the TCP/IP suite. This trend is now being extended to *Local Area Networks* (LANs) in an effort to reduce the number of

[12] The term "suite" refers to a layered architecture in which a layer can be implemented without all of the underlying layers. For example the *traceroute* application is implemented directly on top of the IP layer — the Transport-layer does not have to be present. On the other hand, the term "stack" refers to a layered system in which each layer (except the lowest) requires the one directly below it to be present. For example, an OSI X.400 electronic mail application requires the presence of a full OSI protocol stack.

protocol conversions that data must be put through, especially when connecting heterogeneous LANs over the Internet. In fact, the OSI standard did not originally address the issue of internetworking[13], and was only added as a sub-layer to the OSI Network-layer (layer 3) [Stallings, 1991].

Even though the OSI protocol stack has been criticised for its complexity, it still remains a valuable descriptive tool. It is used later in Chapter 5 to describe the various firewall components and how they relate to each other.

2.3 Internet Protocol (IP)

The IP is an unreliable, connectionless datagram service. There are no guarantees that a given IP datagram will reach its destination. In fact, there is no guarantee that the datagram received is the same as was sent. The official IP specification can be found in RFC 791 [Postel, 1981b]. The basic format of an IP datagram is shown in Figure 2-3.

An IP datagram can be a maximum of 65535 (2^{16}-1) bytes long — it is limited by the two-byte datagram length field in the IP header. In practice, few datagrams of this size are sent because most Link-layer protocols support physical frame lengths of a few thousand bytes only. An IP datagram is split into a number of smaller datagrams if it is too large for the underlying Link-layer, this process is known as *fragmentation*. The reconstruction of fragments at the destination host is known as *re-assembly*. The largest amount of encapsulated data a network interface can transmit is called the *maximum transmission unit* (MTU). For example, Ethernet supports an MTU of 1500 bytes.

For IP datagrams with a destination that is located on the same network as the sender the MTU will already be known. This is due to the fact that the MTU is a parameter that is a part of every NIC specification. The Transport-layer protocol can use the default MTU parameter to limit the size of the message it passes to the Network-layer, therefore, under such circumstances an IP datagram will never be fragmented. However, when a Transport-layer protocol builds a message destined for a host on a different network, it has no way of knowing the route, nor the MTU of each physical network the datagram will traverse before reaching its destination. In this case a default MTU of 576 bytes is used which supports a 512 byte message, a 20 byte TCP header, and a 20 byte IP header. Most Link-layer protocols support an MTU of at least 576 bytes.

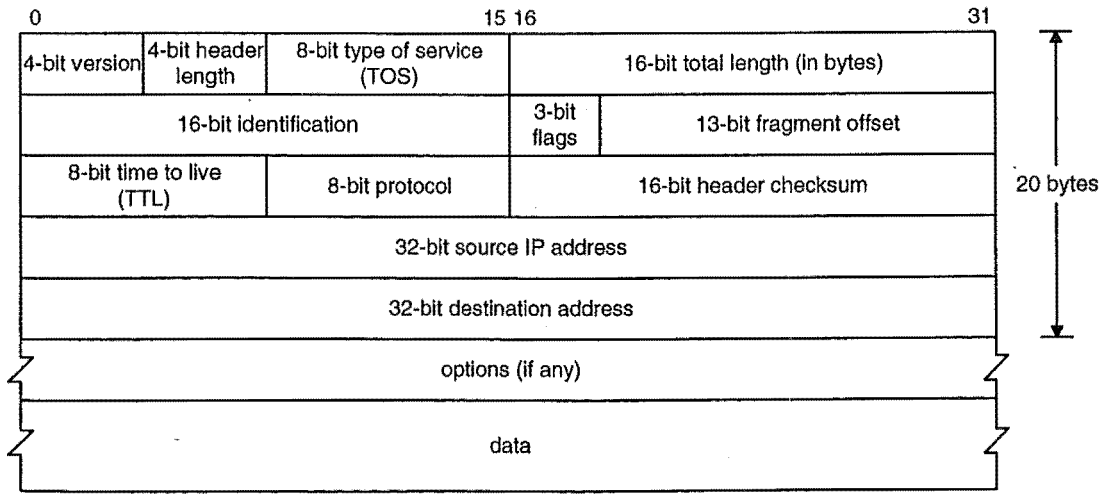

Figure 2-3 IP datagram format.

[13] Internetworking is defined as two communicating end systems not connected to the same network. Therefore data must traverse at least two networks in which the protocol stacks may be quite different.

The following is a brief description, adapted from [Carl-Mitchell et al., 1993], for each of the IP datagram fields shown in Figure 2-3.

- *Version (4-bits)* – the version number of IP; currently version 4 (IPv4) is deployed.
- *Header Length (4-bits)* – the length of the IP header in 32-bit words. The header is always padded out to a multiple of 32-bit words.
- *Type Of Service (TOS) (8-bits)* – the type of service or priority for this datagram. Type of service processing is not frequently used, therefore the default value of 0 is generally used.
- *Total Length (16-bits)* – the length of the IP datagram (including the header) in bytes.
- *Identification (16-bits), Flags (3-bits), and Fragment Offset (13-bits)* – used for fragmentation and re-assembly control.
- *Time To Live (TTL) (8-bits)* – the maximum time in seconds, time-to-live, that the datagram may exist. This field is decremented by at least 1 each time the IP header is processed by a router or host. Unless the datagram is queued in a buffer for a long period of time, this field actually indicates the maximum number of intermediate routers a datagram may cross before it is dropped. When this field reaches 0, it must be dropped unconditionally by the IP. This feature prevents a datagram from looping around the network forever because of a routing error.
- *Protocol (8-bits)* – indicates the type of protocol message encapsulated within the IP header. For example, the protocol field value is 6 for TCP and 17 for UDP.
- *Header Checksum (16-bits)* – provides a checksum for the IP header only. The checksum is constructed by taking the 16-bit 1's complement of all the 16-bit words in the header. This field allows the header to be checked for errors which may have occurred in transmission. This is the only error checking that IP does; other than routing errors.
- *Source IP Address (32-bits)* – the IP address of the interface from which the datagram originated.
- *Destination IP Address (32-bits)* – the IP address of the datagram's final network interface destination. As each IP datagram contains its source and destination address it can be routed independently to its destination.
- *Options (variable bit length)* – can contain various IP options, although most IP datagrams do not. Options include the following:
  - ◊ *source routing* – enables an IP datagram's route to be specifically controlled. Is used in source-routing attacks.
  - ◊ *route recording* – records the route the datagram takes in the options field.
  - ◊ *time-stamping* – adds a time-stamp by each intermediate router.
  - ◊ *security* – can contain seldom used security options (see Section 2.3.2).
- *Padding (variable bit length)* – pads the IP header to an even 4-byte boundary. This is occasionally needed because not all IP options are even multiples of 32-bits.

An IP datagram can travel through many routers or hosts before it reaches its destination. On receipt of a datagram the router looks at its destination address and compares this with its routing table; returning

a result which decides which port the datagram will be sent out on. Routing tables are constantly updated to reflect the status of the various interconnected networks. It is not uncommon for IP datagrams which are part of the same connection to take different paths before arriving at their destination. It is the job of higher layers of the protocol suite, e.g. TCP, to reassemble and re-sequence application data.

Unfortunately, the use of dynamic paths between source and destination points, and the ease at which they can be manipulated, means that any plain text sent across the Internet is, in essence, available for anyone to see.

2.3.1 IP Addresses

As shown in Figure 2-3, IP addresses are 32-bits long and divided into two parts; the network and the host address. The boundary is dependant on the first one to four high-order bits, and indicates which network addressing scheme is being used, as shown in Table 2-1.

Table 2-1 IP Address Formats.

Network Class	High-order bits	Network	Host	Number of Addresses
A	0	7	24	16,777,214
B	10	14	16	65,534
C	110	21	8	254
D	1110	Multicast group		268,435,456
E	1111	(Experimental use)		n/a

The host part of the IP address is usually broken into a subnet and host address. Subnets are used to route IP datagrams within an organisation's network domain. It is up to the organisation to determine the number of bits used for the subnet. For example it is common to divide a Class B network into 254 sub-networks.

IP addresses are not usually used in their numeric formats, instead they are translated into a more human readable form, for example 132.181.10.25 is translated to www.cosc.canterbury.ac.nz. This translation is accomplished by the *Domain Name System* (DNS) (see Section 2.8) which is essentially a distributed database.

There has been a lot of concern recently on the consumption of IP addresses, mainly due to the wastage caused by subnet partitioning. For example, if an organisation has a single class C network address, they have a possible 255 host addresses. However they may only have 50 hosts on their network, thus 205 host addresses have been wasted. If it were not for subnets, the current 32-bit IP addressing scheme could accommodate a possible 2^{32} host addresses. This problem has been addressed in the *IP version 6* (IPv6) standard, by using 128-bit addresses.

It should be noted that IP source addresses do not provide a reliable indication as to the originator of the datagram. In fact, this is another weakness of IPv4 and forms the basis for spoofing attacks discussed in Section 3.2.2.

2.3.2 IP Security Labels

The IP header provides space for a number of optional fields that are not commonly used. The important ones from the perspective of security are the security label and strict and loose source routing (see page 30)

The IP security option [Housley, 1993] [Kent, 1991] is seldom (if ever) used within commercial organisations, but has found significantly more success within military environments. Each datagram is labelled in accordance with the sensitivity of the information that it contains. The labels are designed for compartmentalised multi-level secure (MLS) operating systems, therefore they include both a hierarchical component (e.g. SECRET, TOP SECRET, etc.) and an optional *category* (e.g. nuclear weapons, cryptography, NATO, etc.)

Essentially, the labels are used to indicate the security level of the ultimate sending and receiving processes. A process may not write to a medium with a lower security level, because that would allow the disclosure of confidential information. For obvious reasons, it may not read from a medium containing information more highly classified. Under normal situations, the combination of these two restrictions will usually dictate that the processes at either end of a connection be at the same level. Additional information can be found in [Amoroso, 1994]

Some operating systems, such as UNIX SCO CompartMented Workstation (CMW) / MLS, maintain security labels for each process. This allows them to attach the appropriate option field to each datagram. For conventional computers, a router can attach the option to all packets received on a given connection.

Within the network the primary purpose of security labels is to constrain routing decisions. A datagram marked "TOP SECRET" may not be transmitted over a connection with a lower security level (e.g. CLASSIFIED). An additional use of security labels is to control encryption equipment, for instance the previous described "TOP SECRET" datagram may be routed over an insecure connection provided it has been suitably encrypted with a cryptographic algorithm (see Section 7.3) rated for protection of TOP SECRET messages.

2.4 Internet Control Message Protocol (ICMP)

The Internet Control Message Protocol (ICMP), RFC 792 [Postel, 1981a], is an integral part of the IP suite. It is used to communicate information and control messages between IP hosts. ICMP messages are sent in several situations; for example, when a datagram cannot reach its destination, when a router does not have the buffering capacity to forward a datagram, and when a router can direct the host to send traffic on a shorter route. ICMP messages are sent like other transport protocol messages, that is, they are encapsulated in an IP datagram. The structure of an ICMP message is shown below in Figure 2-4. There are 15 different values for the *type* field, which are used to identify the type of ICMP

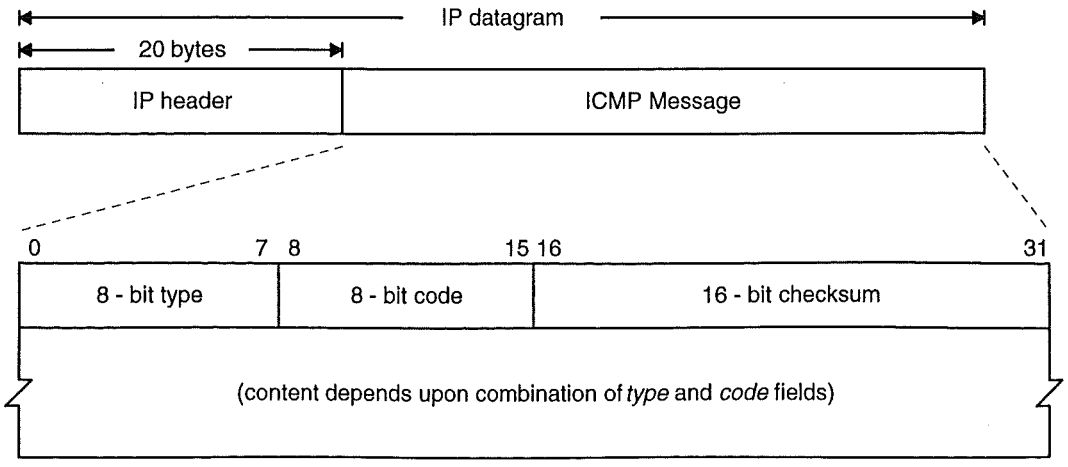

Figure 2-4 ICMP message structure in relation to IP datagram encapsulation.

message (see Table 2-2). Some ICMP messages make use of the *code* field, which provides more specific information on the connections status.

Table 2-2 ICMP message types.

type field	ICMP message *type*
0	echo reply (used by *Ping*)
3	destination unreachable
4	source quench
5	redirect (alter route)
8	echo request (used by *Ping*)
9	router advertisement
10	router solicitation
11	time exceeded for a datagram
12	parameter problem with datagram
13	timestamp request
14	timestamp reply
15	information request (obsolete)
16	information reply (obsolete)
17	address mask request
18	address mask reply

The ICMP protocol also forms the basis of a useful program called "Ping", which stands for *Packet InterNet Groper* [Stevens, 1994]. Ping issues an ICMP echo request message to a host, and expects an ICMP echo reply in return. The lack of a reply indicates that there could be a problem with the destination host or the intervening network. However, many firewalls are configured to block ICMP messages, therefore Pings to these hosts will not receive replies.

The most obvious exploitation of ICMP is through the *redirect* message [Bellovin, 1989] which is a control message used by gateways to advise of better routes. Fortunately, the possible impact from this threat is limited by the constraints placed on the ICMP redirect message;

- must be related to a specific, existing connection,
- cannot be used to make unsolicited changes to a hosts routing table, and
- are only applicable to a limited topology; i.e. they may only be sent from the first gateway along the path to the originating host. A later gateway may not advise that host, nor may it use ICMP redirect messages to control other gateways.

However, if a secondary gateway can be penetrated on a target host's local network (although it may be sufficient to compromise an ordinary host and have it act as a gateway), then it is possible to attack the target host's routing table — redirecting all traffic to the compromised host or secondary gateway. The attack is simplified if hosts do not perform sufficient validation checks on the redirect messages.

Another popular abuse of ICMP is the generation of denial-of-service attacks. It is possible to use several types of ICMP messages, such as *Destination Unreachable* and *Time to Live Exceeded*, to reset existing connections. Some older implementations of ICMP do no limit their action to a specific connection, but will tear down all connections between the host and gateway on receipt of these messages. Some operating systems are vulnerable to a denial-of-service attack because their implementation of Ping cannot deal with oversized ICMP echo request messages (details about this vulnerability can be found in Section 3.2.4).

2.5 Address Resolution Protocol (ARP)

In most cases IP datagrams are sent over data links such as Ethernet or Token Ring. However, these devices have their own addressing schemes, e.g. 48-bits in the case of Ethernet. When Ethernet frames are sent between hosts on a LAN, it is the 48-bit Ethernet address that determines which interface (i.e. NIC) will receive it.

The *Address Resolution Protocol* (ARP), RFC 826 [Plummer, 1982], is used to provide dynamic mapping between the 32-bit IP addresses and the 48-bit Ethernet addresses. Mappings are stored in the hosts ARP cache which is essentially a table where each entry associates an IP address to an Ethernet address. Mappings in the ARP cache are expired after twenty 20 minutes.

If a mapping in the host's ARP cache has expired, or the required one is not found, the host sends an ARP request (contained in an Ethernet broadcast frame) which contains the desired IP address. The destination host (if it exists) sends an ARP reply containing the IP and Ethernet address pair, which is placed by the sending host into its ARP cache. The ARP cache is necessary to reduce the amount of Ethernet broadcast traffic that would be required if IP to Ethernet mappings were resolved through ARP requests only.

It should be noted that an untrusted host which has access to the LAN could broadcast phoney ARP messages to redirect all traffic to itself. The attacker could then impersonate another host and modify its data streams.

2.6 Transmission Control Protocol (TCP)

The TCP, RFC 793 [Postel, 1981d], is a connection oriented protocol that provides end-to-end reliability and in-order sequencing of a byte stream, and attempts to optimise network bandwidth by managing the data flow between the sender and receiver. The TCP protocol processes a stream of data from an application and splits it into a series of segments (or messages) that are passed to the Network-layer for delivery to an application at the other end of the connection. The remote TCP receives the segments from the Network-layer and orders them to recreate the original byte stream, which is passed to the Application-layer. Figure 2-5 shows the TCP segment format.

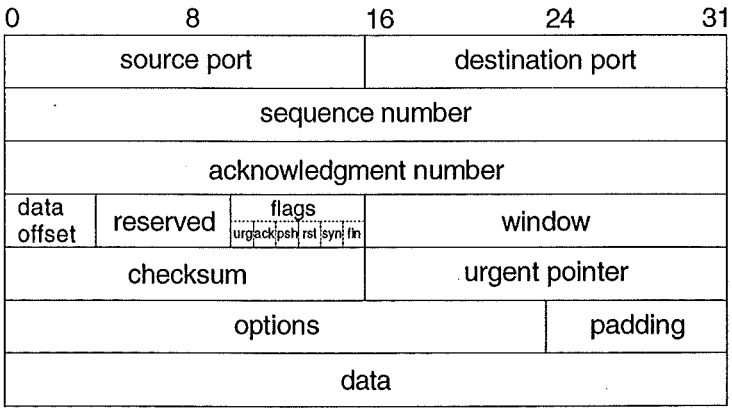

Figure 2-5 TCP segment format.

The following points provide brief descriptions, adapted from [Postel, 1981d], of each TCP segment field shown in Figure 2-5:

- *Source Port (32–bits)* –The source port number.
- *Destination Port (16–bits)* – The destination port number.

- *Sequence Number (32-bits)* – The *sequence number field* (SNF) contains the position of the first data octet in the segment. If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

- *Acknowledgement Number (32–bits)* – If the ACK control flag is set the *acknowledgement number field* (ANF) contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

- *Data Offset (4–bits)* – The number of 32-bit words in the TCP header. This indicates where the data begins. The TCP header (even one including options) is an integral number of 32-bits long.

- *Reserved (6–bits)* – Reserved for future use. Must be zero.

- *Control Bits (6–bits)* – From left to right (see Table 2-3 for a description of each flag);
    1. URG – Urgent
    2. ACK – Acknowledgement
    3. PSH – Push
    4. RST – Reset
    5. SYN – Synchronise
    6. FIN – Finish

- *Window (16–bits)* – The number of data octets beginning with the one indicated in the acknowledgement field which the sender of this segment is willing to accept.

Table 2-3 TCP control flags.

Flag	Description
SYN	*Synchronise Sequence Numbers* – Indicates that the sequence number field contains the connection-initiator's initial sequence number, and is only valid during the 3-way handshake used to initialise a TCP connection. TCP sequence numbers can be thought of as 32-bit counters. They range from 0 to 2^{32}-1. Every byte of data exchanged across a TCP connection (along with certain flags) is sequenced.
ACK	*Acknowledgement* – The acknowledgement number field is almost always set. It indicates that the acknowledgement field of this segment specifies the next sequence number the sender of this segment is expecting to receive, hence acknowledging receipt of all previous sequence numbers.
RST	*Reset* – Indicates that the receiver should delete the connection without further interaction. The receiver can determine, based on the sequence number and acknowledgement fields of the incoming segment, whether it should honour the reset command or ignore it — unless the segment has been spoofed (see Section 3.2). In no case does receipt of a segment containing RST give rise to a RST in response.
URG	*Urgent* – Provides TCP with a way of implementing out of band (OOB) data. For instance, in a Telnet connection a `ctrl-c` on the client side is considered urgent and will cause this flag to be set.
PSH	*Push* – The receiving TCP should not queue this data, but rather pass it to the application as soon as possible. This flag is most often set in interactive connections, such as Telnet and rlogin.
FIN	*Finish* – Indicates that the sending TCP has finished transmitting data, but will still accept data.

- *Checksum (16–bits)* – The checksum field is the 16-bit one's complement of the one's complement sum of all 16-bit words in the header and data fields. If a segment contains an odd number of header and data octets, the last octet is padded on the right with zeros to form a 16-bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros. The checksum also covers a 96-bit pseudo header conceptually prefixed to the TCP header. This pseudo header contains the source IP address, the destination IP address, the protocol, and TCP length. This gives the TCP protection against misrouted segments

- *Urgent Pointer (16–bits)* – This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only interpreted in segments with the URG control bit set.

- *Options (variable bit length)* – Options may occupy space at the end of the TCP header and are a multiple of 8-bits in length. All options are included in the checksum. An option may begin on any octet boundary.

- *Maximum Segment Size Option Data (16–bits)* – If this option is present, then it communicates the maximum receive segment size at the TCP which sends the segment. This field must only be sent in the initial connection request (i.e. in segments with the SYN control bit set). If this option is not used, any segment size is allowed.

- *Padding (variable bit length)* – The TCP header padding is used to ensure that the TCP header ends and data begins on a 32-bit boundary. The padding is composed of zeros.

TCP supports the multiplexing of multiple circuits over a single channel. Every TCP segment includes the originating host address (orig.host) and port number (orig.port), as well as the destination host address (dest.host) and port number (dest.port). This vector (or 4-tuple), <orig.host, orig.port, dest.host, dest.port>, uniquely identifies the circuit being used for the communication.

Communication over the Internet usually conforms to the *Client/Server model.* Servers generally listen to ports numbered below 1024, which are referred to as "well known ports" (see Appendix C). These ports offer standard TCP/IP services such as, Telnet, FTP, SMTP, etc. A Server continually listens to its associated port waiting for a client process to initiate a connection. Port numbers for client processes are generally allocated "high", i.e. above 1023. However, it is unwise to trust services based on port number alone, as the allocation of port numbers is a convention only!

When two applications wish to communicate, their TCP's must first establish a connection across potentially unreliable networks. To provide reliable and error free communication the TCP protocol retransmits lost or damaged segments. This is possible because each segment contains a unique sequence number to identify its position within the original transmission sequence, and a 16-bit checksum to ensure the contents have not changed in transit. All segments contain a *sequence number*, and (except for the first segment used to initiate the connection) an *acknowledgement number* that corresponds to the sequence number of the last successfully received segment (i.e. *segment number + 1*). Sequence numbers also prevent segments that get delayed in the network from being delivered late and being misinterpreted as part of the existing connection. Figure 2-6, page 19, depicts the 3-way handshake between a client (host A) and a server (host B).

The 3 step process shown in Figure 2-6 can be summarised as follows;

- Step 1 – The client (host A) sends a TCP segment, with the SYN flag set, to the server (host B) to indicate the clients intention to establish a connection (this is the only time it is valid to set the SYN flag). At this point the SNF, SNF_A, contains the ISN, ISN_A, which is usually generated as a function of the time-of-day clock.

- Step 2 – On receipt of the clients TCP segment the server responds with its own segment. The server's ISN, ISN_B, is assigned to the SNF, SNF_B, and the SYN flag is set. The server must also

acknowledge the clients segment, so the ACK flag is set and the ANF, ANF_B, is assigned the value of $SNF_A + 1$.

- Step 3 – On receipt of the server's SYN/ACK segment the client must respond with an ACK segment to complete the handshake. The client's ACK segment is created by setting the ACK flag, and assigning ANF_A the value of $SNF_B + 1$. From this point forward the client can start sending TCP segments containing application data. The server can do likewise once it has received the client's ACK segment.

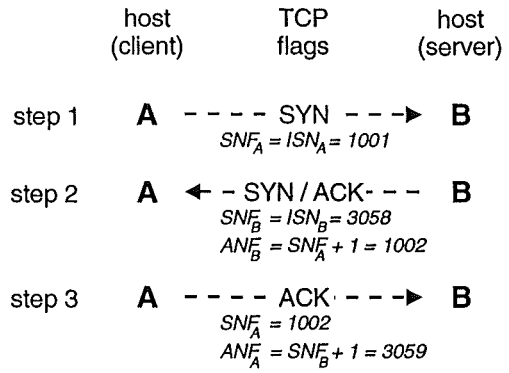

Figure 2-6 TCP 3-way handshake.

Although simple and effective, the three-way handshake presents a vulnerability if an attacker can predict the target's ISN [Morris, 1985] [Bellovin, 1989]. An attacker can then fool the target into thinking that it is communicating with a trusted host. A full description of this and related attacks can be found in Section 3.2.

2.7 User Datagram Protocol (UDP)

The UDP, RFC 768 [Postel, 1980], provides a datagram mode of packet-switched computer communication and assumes that IP is used as the underlying protocol.

UDP provides a procedure for application programs to send messages to other programs with a minimum of protocol overhead. The protocol is connectionless, so delivery and duplication protection is not guaranteed. It is well suited for transaction based processes, such as Sun's Remote Procedure Calls [Sun Microsystems, 1988].

UDP tends to behave badly when used to transmit streams of data because it lacks flow control. Thus, UDP messages can swamp hosts and routers causing extensive datagram loss. This characteristic can be exploited by attackers to launch *denial-of-service* attacks.

It is also far easier to spoof the UDP than the TCP because there are no handshaking protocols, and the datagrams are unique (i.e. there is no notion of sequence number). Thus, it is not recommended that the source address be used for authentication.

2.8 Domain Name Service

The DNS is a distributed database used by TCP/IP applications to map between hostnames and IP addresses, and is also used to determine the destination of email. All hosts connected to the Internet have unique IP addresses, which are used to communicate with one another. However, IP addresses are not easily remembered by humans, so the DNS provides a way of associating an ASCII based identifier to an IP address and mapping between them. No single site on the Internet contains all mapping

information. Instead every Internet site (e.g. University, company, etc.) maintains its own database and runs a server program that other systems (i.e. clients) across the Internet can query. DNS provides the protocol which enables clients and servers to communicate with each other.

Applications access the DNS through *resolvers* which on UNIX hosts are typically the *gethostbyname()* and *gethostbyaddr()* library functions. The first resolver, gethostbyname, takes a host name and returns its IP address. The IP address returned by querying a DNS is not related to the choice of name for a host. The second resolver, gethostbyaddr, takes an IP address and returns the corresponding host name. The resolver may have to contact more than one DNS to complete the mapping.

Each DNS entry has associated with it a number of records which store the information required to respond to resolver queries. The supported DNS record types are shown in Table 2-4.

Table 2-4 DNS record types.

Record Type	Description
A	Authoritative address for IP version 4
AAAA	Authoritative address for IP version 6
NS	Name Server
CNAME	Canonical name for an alias for a hostname
PTR	Pointer record — maps IP address to a hostname
HINFO	Provides host information
MX	Mail exchange record — specifies an alternate computer to receive mail for a particular host
AXFR	Request for zone transfer
ANY	Request for all records

Normally a resolver will generate a UDP based query to the DNS, which replies with the correct mapping, or returns information on an alternative DNS which can be queried further. TCP can also be used for queries, however, this is normally reserved for *zone transfers*. A zone transfer allows backup servers to obtain a full copy of their portion of the DNS name space. This can also be used by attackers to obtain a list of potential targets.

The DNS name space is similar to a hierarchical file system, which is depicted below in Figure 2-7, page 21. Each node (i.e. circle) has a maximum 63 character label, with the exception of the root node which has a null label. Lower and upper case characters are considered equivalent.

At the top are the root name servers, which contain information about the contents of the top level domains (TLD), i.e., .net, .com, .edu, .org, .int, .gov, .mil, and the two-letter country codes from ISO-3166 (.nz, .us, .uk, .fi, .jp, etc.) [Barkow, 1996]. Under the top-level domains are the second level domains, such as nsa.gov. Further levels can be defined, but are the responsibility of the second level DNSs maintained by the organisation, in this case the nsa. Each name reflects its position within the DNS tree, for example, www.nsa.gov represents a host computer (www) in the nsa domain which is under the .gov top level domain. The domain name for a node is constructed by starting at that node and working up the tree to the root, separating each label with a period.

A query about www.nsa.gov sent to a primary domain server will be answered (as long as the primary domain server knows about the domain nsa.gov) with a pointer (IP address) to the nsa.gov name server which holds the information about the host www.nsa.gov. The nsa.gov DNS will then return the IP address for the host www.nsa.gov. At this point a direct connection can be made to www.nsa.gov by the querying host.

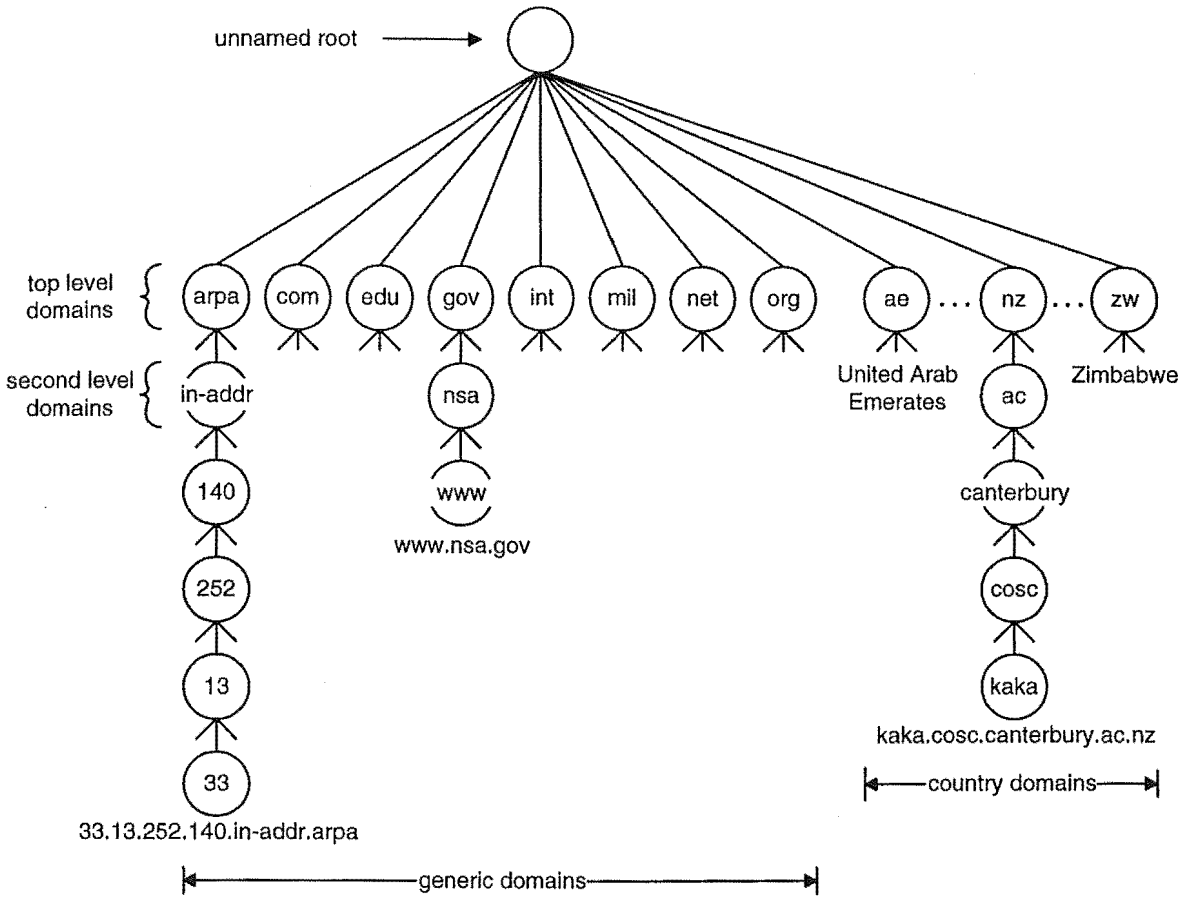

Figure 2-7 Hierarchical organisation of the DNS.

With the emergence of the Internet as a marketing and commercial tool, the TLDs .com, .org, and .net are becoming unwieldy. For this reason the *Internet Assigning Numbers Authority* (IANA), responsible for managing DNS, has been moving new domain registries toward the less used country-code domains.

An attacker who gains control of a DNS is in a very advantageous position because the attacker can send fraudulent information to any host which queries them. For this reason the DNS is generally considered an untrusted service. Hosts which place undue trust in the information returned by a DNS may provide an avenue for an attacker to gain control of their hosts and networks. Threats to the DNS are discussed in Section 2.8.

2.9 Summary

The TCP/IP suite has been evolving since the late 1970s, and has become the world's most predominant WAN protocol. It is even gaining acceptance over traditional LAN protocols, such as Novell's IPX/SPX and Microsoft's NetBEIU, as organisations implement Intranets and Extranets.

The purpose of this Chapter was to introduce the most fundamental protocols of the TCP/IP suite. This is necessary for understanding Internet technologies such as firewalls and VPNs, as well as the threats and vulnerabilities they protect against.

Although the TCP/IP suite has proven to be exceptionally adaptable and resilient, it provides only the most rudimentary support for network security. In particular, IPv4 does not protect the confidentiality of the higher layer data that it transports across the many interconnected networks that comprise the Internet. Throughout this Chapter numerous references have been made to the security problems that

plague the TCP/IP suite at the Network, Transport, and Application-layers. A detailed discussion of the common threats and vulnerabilities which effect the TCP/IP suite is given in Chapter 3.

However, action is underway to provide security mechanisms for IPv4, and its proposed replacement IPv6, in the form of IP layer security, or IPSec. At the Transport and Application-layers similar efforts, including SSL and PPTP, are also underway to provide the necessary security mechanisms. Chapter 7, discusses these protocols in detail, and looks at how the TCP/IP suite can be made secure through their use.

Chapter 3. Threats from the Internet

3.1 Introduction

With such a large number of users, inherently there are those whose motives for using the Internet are not benign. These people are generally referred to as "hackers" or "crackers", terms which tend to glorify and contribute a sense of mystique to their exploits. To avoid such connotations, the remainder of this thesis will use the term "attacker" to emphasise the fact that these people are intentionally launching attacks and invading networks.

Even though there are growing pressures on organisations to connect to the Internet they need to be aware of the risks as well as the benefits. A survey of organisations in the UK [CFS, 1996b] indicated that 65% of respondents with access to the Internet are not even aware of what connections their employees can make. This same survey also revealed that 75% of the organisations did not have a designated security officer, 77% had no formal procedures for reporting security incidents, and 67% did not have a business continuity plan.

The 1998 Computer Crime and Security Survey [CSI, 1998] conducted by the Computer Security Institute (CSI) and Federal Bureau of Investigation (FBI) reports a continuing growth in US computer crime. The survey received responses from 520 security practitioners covering a cross-section of US corporations, government agencies, financial institutions and universities. The following points summarise a number of the most interesting results:

- 64% of respondents reported computer security breaches in the twelve months prior to March 1998. This figure represents an increases of 16% over the "1997 CSI/FBI Computer Crime and Security Survey" results, in which 48% of respondents reported unauthorised use. The 1998 survey represents a 22% increase over the initial 1996 survey, in which 42% acknowledged unauthorised use.

- Although 72% of respondents acknowledge suffering financial losses from such security breaches, only 46% were able to quantify their losses. The total financial loss for the 241 organisations that could quantify a dollar figure was US$136,822,000. This figure represents a 36% increase in reported losses over the 1997 figure of US$100,115,555.

- Security breaches detected by respondents include a diverse array of attacks. For example, 44% reported unauthorised access by employees, 25% reported denial-of-service attacks, 24% reported system penetration from the outside, 18% reported theft of proprietary information, 15% reported incidents of financial fraud, and 14% reported sabotage of data or networks.

- The most serious financial losses occurred through unauthorised access by insiders (18 respondents reported a total of US$50,565,000 in losses), theft of proprietary information (20 respondents reported a total of US$33,545,000 in losses), telecommunications fraud (32 respondents reported a total of US$17,256,000 in losses) and financial fraud (29 respondents reported a total of US$11,239,000 in losses).

- The number of organisations that cited their Internet connection as a frequent point of attack rose from 47% in 1997 to 54% in 1998. This represents a 17% increase over the initial 1996 figure of 37%. Significantly, the number of respondents citing their Internet connection as a frequent point of attack is now equal to the number of respondents citing internal systems as a frequent point of attack. In the past, internal systems have been considered the greatest threat, however, it is not the internal threat that has diminished, but simply that the external threat (e.g. Internet connections) has increased. This conclusion is reinforced by another piece of data; of those acknowledging unauthorised use, 74% reported from one to five incidents originating outside the organisation, while 70% reported from one to five incidents originating inside the organisation.

Table 3-2 reproduced from the 1998 CSI survey classifies the financial losses caused by computer crime into a number of generalised categories (Note: monetary figures are in US dollars).

Table 3-2 The aggregate cost of computer crime and security breaches over a 24-month period (1997 – 1998). (Note: 72% of respondents reported suffering financial losses, however only 42% could quantify the losses.)

	Incidents w/ Quantified Losses			Lowest Reported		Highest Reported		Average Loss		Total Loss		
	1997	1998	97-98	1997	1998	1997	1998	1997	1998	1997	1998	97-98
Theft of proprietary info.	21	20	41	$1,000	$300	$10,000,000	$25,000,000	$954,666	$1,677,000	$20,047,986	$33,540,000	$53,587,986
Sabotage of data or networks	14	25	39	$150	$400	$1,000,000	$500,000	$164,840	$86,000	$2,307,760	$2,150,000	$4,457,760
Telecom eavesdropping	8	10	18	$1,000	$1,000	$100,000	$200,000	$45,423	$56,000	$363,384	$560,000	$923,384
Systems penetration by outsider	22	19	41	$200	$500	$1,500,000	$500,000	$132,250	$86,000	$2,909,500	$1,634,000	$4,543,500
Insider abuse of Net access	55	67	122	$100	$500	$100,000	$1,000,000	$18,304	$56,000	$1,006,720	$3,752,000	$4,758,720
Financial fraud	26	29	55	$5,000	$1,000	$2,000,000	$2,000,000	$957,384	$388,000	$24,891,984	$11,252,000	$36,143,984
Denial-of-service	n/a	36	36	n/a	$200	n/a	$1,000,000		$77,000	n/a	$2,772,000	$2,772,000
Spoofing	4	n/a	4	$1,000	n/a	$500,000	n/a	$128,000	n/a	$512,000	n/a	$512,000
Virus	165	143	308	$100	$50	$500,000	$2,000,000	$75,746	$55,000	$12,498,090	$7,865,000	$20,363,090
Unauthorised insider access	22	18	40	$100	$1,000	$1,200,000	$50,000,000	$181,437	$2,809,000	$3,991,614	$50,562,000	$54,553,614
Telecom fraud	35	32	67	$300	$500	$12,000,000	$15,000,000	$647,437	$539,000	$22,660,295	$17,248,000	$39,908,295
Active wiretapping	n/a	5	5	n/a	$30,000	n/a	$100,000		$49,000	n/a	$245,000	$245,000
Laptop theft	160	162	322	$1,000	$1,000	$1,000,000	$500,000	$38,326	$32,000	$6,132,160	$5,184,000	$11,316,160
									Total	$97,321,493	$136,764,000	$234,085,493

Source: CSI/FBI 1998 Computer Crime Survey

The participants of this survey were also asked to indicate the types of security technology that they were using, the results are shown in Table 3-1. Of the 512 respondents it is interesting to note that 81% have invested in firewall technology, unfortunately it is not clear whether the remaining 19% use

Table 3-1 Types of security technology in use by respondents to the 1998 CSI/FBI Computer Crime and Security survey.

Security Technology	Implemented by
Access control	89%
Biometrics	6%
Encrypted Files	49%
Anti-virus software	96%
Reusable passwords	53%
Firewalls	81%
Encrypted login	36%
Physical security	89%
PCMCIA	34%
Intrusion detection	35%
Digital IDs	20%

alternative methods to protect their Internet connections, are not connected to the Internet, or simply take no precautions. It is also interesting that cryptography is being used to protect files, and logins, but there is no indication that VPNs are becoming a core security technology.

Although the statistics presented above are helpful when analysing attack trends and for determining where threats lie, they do not provide an understanding of the attacks themselves. It is essential that attack methods are understood, especially when developing security policies and procedures to address the right problems in the most effective and efficient way.

The following points introduce a few of the most common attacks [DeMaio, 1995] used by both internal and external attackers:

- *Password Guessing* – It is relatively easy to obtain a password cracking program such as "Crack"[14]. These programs use standard and non standard dictionaries, and simply try to guess an account's password. Usually, such programs find at least 10 percent of the passwords chosen by users [DeMaio, 1995]. Educating users as to the correct selection and use of passwords is the most effective solution.

- *Password Sniffing* – The *Computer Emergency Response Team* (CERT) Co-ordination Centre estimates that in 1994 more than 100,000 systems were the victim of password sniffers. Once a hacker gains entry to a system they will usually install sniffer programs to automatically capture passwords and account information. The programs typically monitor TCP sessions, such as Telnet or FTP, and record the first 128 or so bytes that contain the identification and authentication information. The best defence is to employ one-time password schemes.

- *IP Spoofing* – There are a number of spoofing attacks which take advantage of the information contained within the IP datagram header. For example the Christmas Day attack on Tsutomu Shimomura[15] involved forging the source address of the IP datagrams so they looked as if they were generated from within Shimomura's network. Another IP attack involves loose source routing of IP datagrams (see Section 3.2.2). The attacker manipulates the IP header's source routing option to change the path that datagrams should take. Properly configured firewalls capable of packet filtering provide the best means of defence against these types of attack.

Some attacks such as password sniffing and spoofing are much easier to conduct for an attacker who is internal to the organisation. This is because some attacks require access to the network traffic, it is difficult (even impossible) to do this on the Internet unless the attacker controls a routing node. Although these attacks are only practical due to the presence of a vulnerability, it is important to consider the reasons behind their launch so the risks can be fully understood. The following points describe a few of the reasons an attack may be launched:

- *Information Theft* – Using knowledge of certain Internet services, such as NFS (see Section 3.3.5), attackers can spoof the host authentication mechanism to gain access to sensitive information. If an attacker has access to a valid password, then depending on the level of access it provides, sensitive organisational information and data may be at risk. Computer Weekly reported findings by the CSI which indicated that information theft rose 260% from 1985 to 1993. Of 8,932 attacks, 7,860 were successful, but only 19 were reported [Computer Weekly, 1995]. A properly configured firewall can help prevent unauthorised access by providing countermeasures such as strong authentication or network encryption.

[14] "Crack" is available at **ftp://ftp.cert.org/pub/tools/crack**

[15] On Christmas Day, 1994, a hacker launched a sophisticated "IP spoofing" attack against the home computer of a computer security expert, Tsutomu Shimomura, a researcher at the federally financed San Diego Supercomputer Center in California. Over a two week period Shimomura pursued and eventually tracked the hacker to computers on which Shimomura's stolen files were found. The hacker was finally identified as Federal fugitive Kevin Mitnick, and subsequently arrested at 1:30 a.m. on February 15, 1995 by FBI agents. The following WWW-addresses provide links to a great deal of interesting information regarding the Christmas Day attack and Kevin Mitnick; **http://www.gulker.com/ra/hack/** and **http://www.mitnick.com/**

- *Denial-of-service* – This is a class of attack designed to prevent the use of computers and networks by legitimate users. Some attacks, such as "Ping O' Death" (see Section 3.2.4), can completely shut down or disable equipment and services. For example, it is possible to send ICMP redirect messages to a host or router telling it to stop sending IP datagrams to all or part of a network. More common are flooding attacks, e.g. SYN flooding (see Section 3.2.1), which overload a computer or network so that it spends all of its time responding to illegitimate messages and requests. Solutions include placing services on separate hosts so if one is flooded the others will continue to function, or using a properly configured firewall to filter out dangerous protocols, such as ICMP redirect messages.

- *Information Destruction* – In some cases an attacker's intentions may be purely malicious, the aim of their attack being the destruction of an organisations information. Related to this is unauthorised data modification, for example an attacker may wish to confuse experimental results, or alter the number of units sent to a customer. Obviously, the reasons behind such attacks are complex, they may be driven by revenge, corporate rivalry, or just plain delinquency. It is easy to think of many situations in which an organisation could suffer such attacks.

The remainder of this Chapter looks at the vulnerabilities exploited by attackers, and focuses on the TCP/IP suite, and on the applications that overlay it.

3.2 Threats to the TCP/IP Protocol

This Section describes a number of common attacks which exploit the limitations and inherent vulnerabilities in the TCP and the IP. The following attacks are discussed in detail:

- SYN flooding
- IP Spoofing
- Sequence number attack
- TCP session hijacking
- RST and FIN denial-of-service attack
- Ping of Death

These attacks were chosen because software to launch them (including source code) is freely available on the Internet. They are also the most common, and practical attacks used by attackers on the inside and outside of organisations networks.

3.2.1 SYN Flooding

Description

SYN flooding occurs when a server receives more incomplete connection requests than it can handle [Stewart et al., 1997]. In 1996 both *2600*[16] and *Phrack*[17], two of the largest and most well-known of the underground hacker magazines, released source code that automated this attack.

[16] The 2600 WWW-site is available at **http://www.2600.com/**

[17] The Phrack WWW-site is available at **http://www.phrack.com/**

Under normal conditions hosts that wish to exchange data over a TCP connection must initiate the session using a 3 step process known as the *3-way handshake* (see Section 2.6, page 16). The SYN flood attack is based on preventing the completion of the 3-way handshake — in particular the server's reception of the TCP ACK flag (see Table 2-3, page 17).

Unlike a normal TCP connection request, the SYN flood attack withholds the final ACK segment which leaves a server's port in a half-open state. The attack succeeds because the number of half-open connections that can be supported per TCP port is limited. When the number of half-open connections is exceeded the server rejects all subsequent incoming connection requests until the existing requests time out, usually after 75 seconds — creating a denial-of-service condition.

To initiate the SYN flood attack, the attacking host sends a number of SYN requests to the target TCP port (e.g. the Telnet daemon) to fill up its concurrent connection request (or *backlog*) queue — the exact number depends upon the operating system [Phrack, 1996a]. The backlog queue allows a server (i.e. listening port) to queue concurrent connection requests for later processing. To achieve this the details of each pending connection request are stored in a memory structure. Obviously, this queue must be bounded otherwise an attacker could make unlimited connection requests to a TCP port and consume all of the server's memory resources — which in itself would constitute a denial-of-service attack!

The attacking host must ensure that the source IP-address is spoofed to be that of a routable but unreachable host, as the target host will be sending its response to this address. IP (by way of ICMP) will inform TCP that the host is unreachable, however, TCP considers these errors to be transient and leaves their resolution up to IP (reroute the datagrams, etc.) — in effect ignoring them. The IP destination address must be unreachable because the attacker does not want *any* host to receive the SYN/ACKs sent by the target host, as this would elicit a RST from that host and defeat the attack.

Figure 3-1 shows the steps involved in launching a TCP SYN flood attack. To begin (step 1) the attacking host sends a multitude of SYN requests to the target to fill its backlog queue with pending connections. Once the target receives this request it responds with SYN/ACKs (step 2) to what it believes is the source of the incoming SYNs. Once the backlog queue is full all further requests to the TCP port will be ignored until the original requests begin to time out and reset — normally after 75 seconds. After each time out (step 3) the server port sends a RST to the unreachable client. At this point the attacker must repeat the process again (from step 1) to continue the denial-of-service attack.

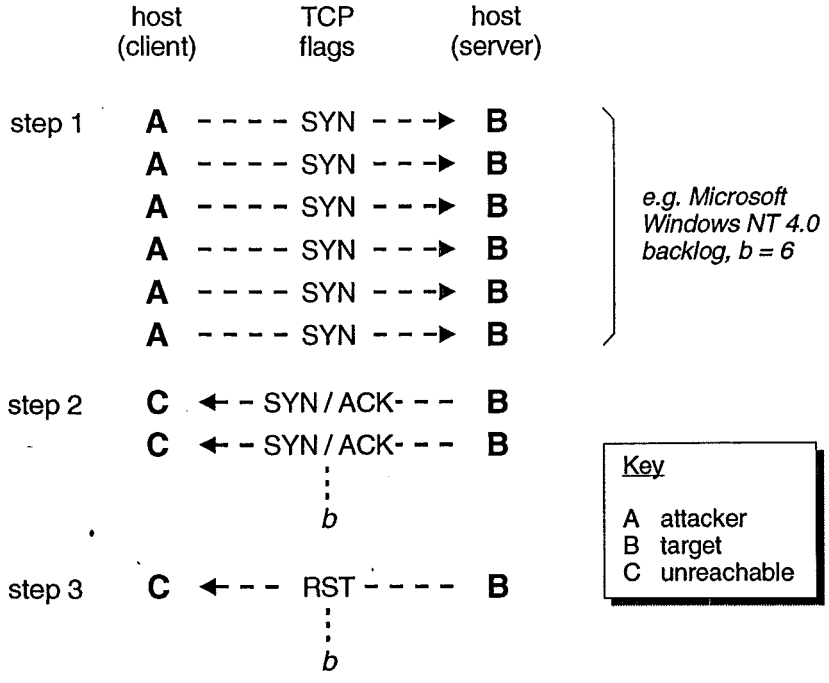

Figure 3-1 TCP SYN flood attack.

To make this attack more difficult to detect and respond to, the software randomises the source address of the IP datagrams sent by the attacking host. Thus, the target host receives datagrams that appear to be from all over the Internet, assuring the attackers anonymity.

Countermeasure

There are several ways of reducing the effectiveness of the SYN flood attack. The first relies on ISPs being responsible enough to block IP datagrams with non-internal addresses from leaving their network and reaching the Internet. Therefore, an attacker would have to send datagrams with an official IP source address which in most cases would lead back (through audit logs) to the owner of the account the attack was being launched from. This lack of anonymity would deter most attackers, although skilled and determined attackers would use accounts that had been compromised or launch attacks through sites that do not regulate Internet traffic.

Other preventative measures require changes to the network aspects of the operating system, or the addition of intrusion detection tools. For example a list of connection requests could be kept with details of the source address, TTL, sequence numbers, windows size etc. These variables could then be analysed for suspicious activity, which if detected, would result in a RST being sent to the half open connection to allow new connections to be made. Other solutions rely on increasing the size of the backlog queue (not particularly effective as the attacker can simply send more SYN segments), and randomly dropping half-open connection requests when the queue is full.

3.2.2 IP Spoofing, TCP Sequence Number Prediction, and TCP Session Hijack

Description

IP spoofing is an attack in which the attacker impersonates a host (or a legitimate user) at the IP layer. In most cases the objective is to attack the trust-relationship between two hosts, which relies upon the source IP address to authenticate hosts. The attack is only possible if the target host has a trust-relationship with at least one other host. The most popular trust-relationship is provided by the *.rhosts* file found on UNIX operating systems, although many others exist, e.g. the UNIX files *hosts.allow*, *hosts.equiv*, etc. The rhosts file allows a user to build a set of trusted hosts applicable only to themselves. For example, suppose that the *~ray/.rhosts* file on the host *huia.canterbury.ac.nz* contained the lines:

```
kaka.canterbury.ac.nz
matata.canterbury.ac.nz
```

This rhosts file would allow an account named *ray* on *kaka* or on *matata* to *rlogin* into *ray's* account on *huia* without typing a password!

In itself IP spoofing is quite simple, all the attacker has to do is generate an IP datagram with a forged source address. This is usually done by creating an IP datagram from scratch using RAW-Sockets[18]. The target host has no way of determining that an IP datagram has been spoofed, as all it has to rely on is the IP source address. On its own IP spoofing is limited to providing anonymity for an attacker launching attacks against the IP layer, e.g. SYN flooding, ICMP redirects, Ping flooding, etc. Therefore, to complete the attack against trust-relationships as described above, the attacker must combine IP spoofing with TCP sequence number prediction — providing the attacker with a delivery mechanism for sending application data to the target host.

[18] The term "Raw-Sockets" refers to the ability in 4.2BSD derived socket implementations to access the Network-layer instead of the Transport-layer. For example, the programmer could directly format the fields within the IP datagram to generate ICMP echo requests (i.e. Ping).

TCP sequence number prediction is used by attackers to attack TCP sessions, and takes advantage of the fact that TCP is a sequenced data delivery protocol (see Section 2.6, page 16). TCP segments are encapsulated within IP datagrams, because of this there is no guarantee that the datagrams will follow the same route and therefore arrive in the order they were sent, in addition network errors may require datagrams to be resent. The TCP protocol uses sequence numbers to ensure that the Application-layer receives data in the same order that it was sent. Although this is a simple and effective method of ensuring a sequenced data stream, it unfortunately introduces a vulnerability. If an attacker can guess the correct sequence number they can then generate their own TCP segments that will be accepted by the target host's TCP layer.

There are really two variations on this attack depending on how early the TCP session is attacked. The first three steps in Figure 3-2 shows the normal TCP 3-way handshake, if successful both the client and server proceed to step 4 and may begin exchanging data. The attacker can choose to attack the TCP handshake to take advantage of a trust relationship — often referred to as IP spoofing but to avoid ambiguity will be known here as *TCP spoofing*; or can wait until step 4 to take over a legitimate TCP session — referred to as *TCP session hijacking*. Full technical discussions regarding TCP sequence number prediction and TCP session hijacking[19] can be found in [Bellovin, 1989] [Morris, 1985] [Joncheray, 1995], and [Phrack, 1996b].

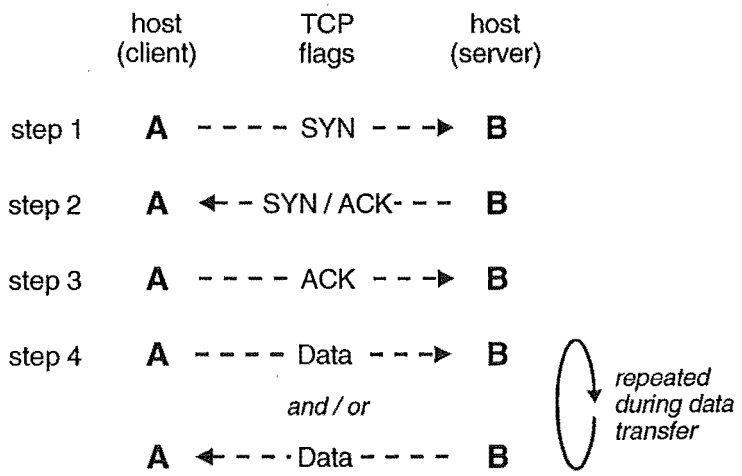

Figure 3-2 TCP 3-way handshake and data transfer.

There are two ways to carry out TCP spoofing attacks;

- *Non-Blind Spoofing* – In this case the attacker is on the same network path as the spoofed host or the target host (e.g. Ethernet 10Base2 LAN) and has direct access to the IP datagrams which contain the TCP segments. Therefore, sequence number prediction is trivial because the attacker simply uses a protocol analyser to capture the TCP segments and obtain the required sequence number.

- *Blind Spoofing* – Is more difficult because the attacker is not on the same network path as the spoofed host or the target host, therefore direct access to the IP datagrams and TCP segments is not possible. Instead the attacker must attempt to guess the correct initial TCP sequence number, the success of which depends upon the mechanism being used to generate it. There are three mechanisms in common use:

    - ◊ *64K rule* – this is the simplest mechanism and surprisingly is still used, or can be found on hosts running older operating systems (e.g. OSF, SunOS). Most

[19] Example code, known as Spoofit, for hijacking Telnet sessions is available from **http://sniffit.rug.ac.be/**. This site also contains a great deal of information about IP spoofing and sequence number prediction.

spoofing programs still provide support to take advantage of this rule. The rule is implemented as follows:

- increase the initial sequence counter every second with a constant (normally 128,000).

- If there is a connection initiated, increase the sequence counter with another constant (normally 64,000).

Obviously, such a mechanism is very easy to predict, especially as the sequence counter is only altered once per second — a very large period in network time!

◊ *time related generation* – is a very popular and simple mechanism which allows the sequence number generator to generate time dependant values. The number generator is seeded at boot time, and is increased on a regular basis (e.g. *μsec*) by an *x* number of *time-units*. Note that time-units on computers are not necessarily perfect, nor are all time-units of equal length, depending on how they are measured and on the load of the computer, etc. This variability increases the difficulty of predicting a correct sequence number.

◊ *pseudo-random generation* – in an effort to foil the prediction of initial sequence numbers newer operating systems are using pseudo-random number generators to generate the values — which makes prediction nearly impossible.

In both cases the attacker must ensure that the spoofed host is unreachable, otherwise it will receive a SYN/ACK (see step 2 of Figure 3-2, page 29) from the target host in response to the attackers spoofed connection request. However, the spoofed host has no knowledge of initiating a connection request and will send a RST to the target host which will cause it to abort the connection and defeat the attack. The attacker normally has two options to deal with this problem, either to wait until the spoofed host is unreachable because of maintenance, or taking it off-line with a denial-of-service attack such as a SYN flood (see Section 3.2.1). An example of a blind spoofing attack is shown in Figure 3-3, page 31.

From the attackers perspective blind spoofing is difficult because all replies from the target host are sent to the spoofed host. Therefore, the attacker cannot determine directly the success or failure of their attack. However, there are ways for attackers to turn a blind spoof into a non-blind spoof. This is achieved by using source routed IP datagrams [Stevens, 1994], or by directly effecting the routing tables of intermediary gateways and routers. Source routing[20] is a feature (an option) of the IP protocol which allows the sender to specify a route for an IP datagram to follow. The route is recorded in the IP header and the receiver uses the reverse of this to send replies. Therefore, an attacker could send source routed IP datagrams appearing to come from the spoofed host and including a route that sends replies back past the attacker. This is one reason why it is important to drop source routed IP datagrams, especially those originating from untrusted networks.

In addition to source routing it is also possible to change the routing tables of gateways and routers by sending spoofed routing datagrams using protocols, such as, RIP, EGP, BGP, etc. As with source routed IP datagrams it is important to ensure that gateways and routers ignore or respond sensibly to the routing information they receive. In most cases though the Internet routes are stable enough so that all routing datagrams can be ignored.

[20] Source routing allows the sender to specify the route of an IP datagram. Two forms are provided; *strict* and *loose* source routing. Strict source routing allows the sender to specify the exact path that the IP datagram must follow. Loose source routing allows the sender to specify a list of IP addresses that the datagram must traverse, but the datagram can also pass through other routers between any two addresses in the list.

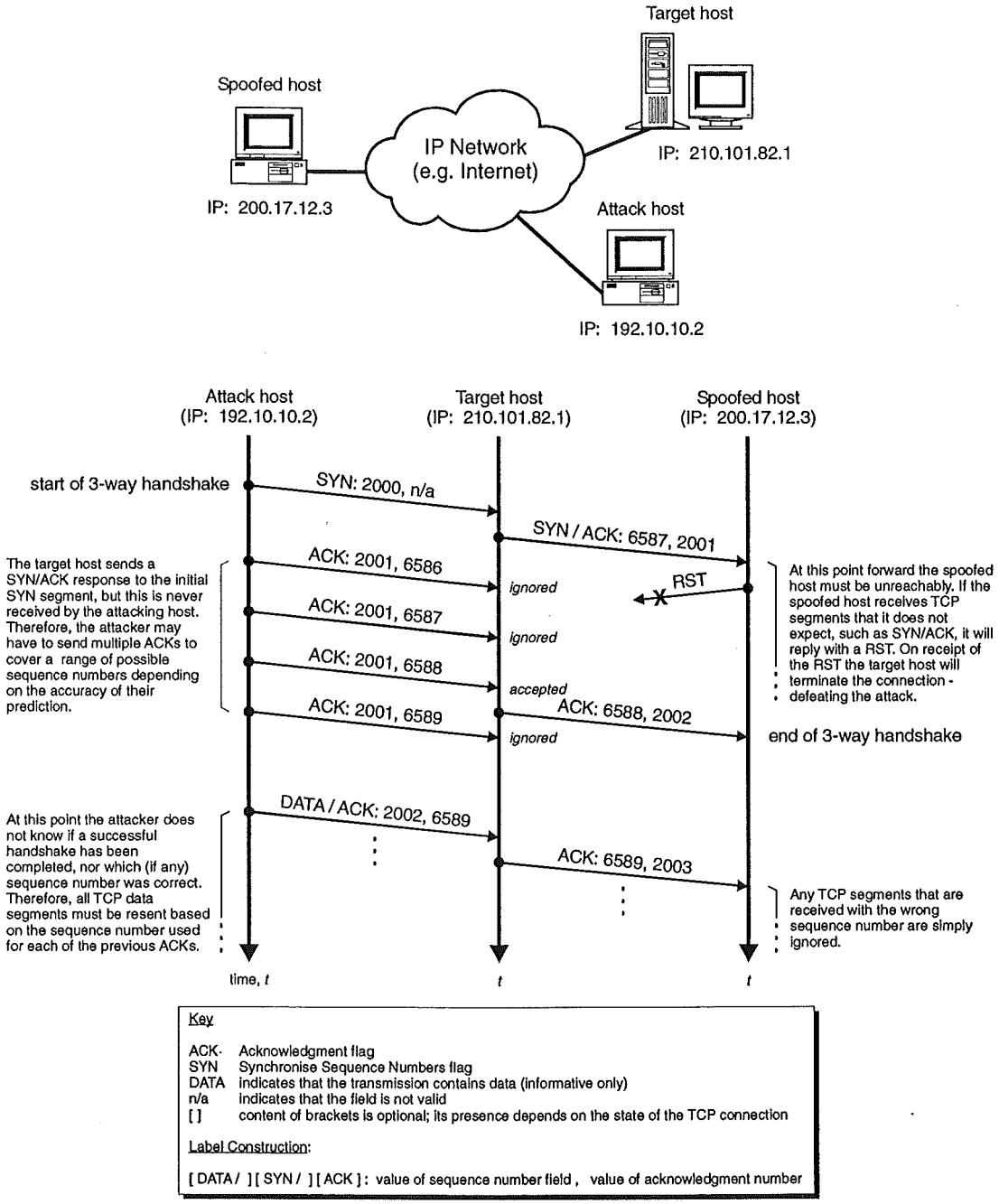

Figure 3-3 Example of a blind spoofing attack.

The final attack, based on IP spoofing and TCP sequence number prediction, is *TCP session hijacking* which can be carried out against any TCP based application, e.g. Telnet, rlogin, FTP, etc. The only requirement is that the attacker has access to the IP datagrams sent between the target and spoofed host, this is necessary to obtaining the correct sequence number. Once the attacker has the correct sequence number a TCP segment can be sent, effectively hijacking the connection — all further datagrams sent by the spoofed host will be ignored by the target host because the sequence numbers will be incorrect. An example of TCP session hijacking is shown in Figure 3-4, page 32.

Generally, TCP hijacking is used to take over a Telnet session. Telnet is a particularly easy protocol to hijack because it simply passes a stream of bytes between the client and server. All the attacker has to do is insert their commands (as a sequence of bytes) into the spoofed TCP data segments. The server will reassemble the TCP segments into command strings which will then be executed as though the

legitimate user had typed them. The only evidence of this attack is that the legitimate user's Telnet session hangs because it never receives conformation of the segments it sends, and will simply continue to resend them. After a few seconds the user will probably attribute the inactivity to "Murphy's Law" and begin a new session.

TCP session hijacking has a number of benefits over other attacks, such as sniffing IP datagrams for passwords, especially when advanced identification and authentication techniques are in use. For example it is pointless to sniff one-time passwords, or responses to challenges issued by cryptographic authentication mechanisms, e.g. S/Key, SecureID, Lockout, etc. However, because all of these advanced authentication techniques happen at connection time, no protection is afforded by them after this point. Therefore, the attacker simply hijacks a legitimate connection to gain entry to a system, and has the added advantage of appearing to the operating system's security mechanisms as the legitimate user!

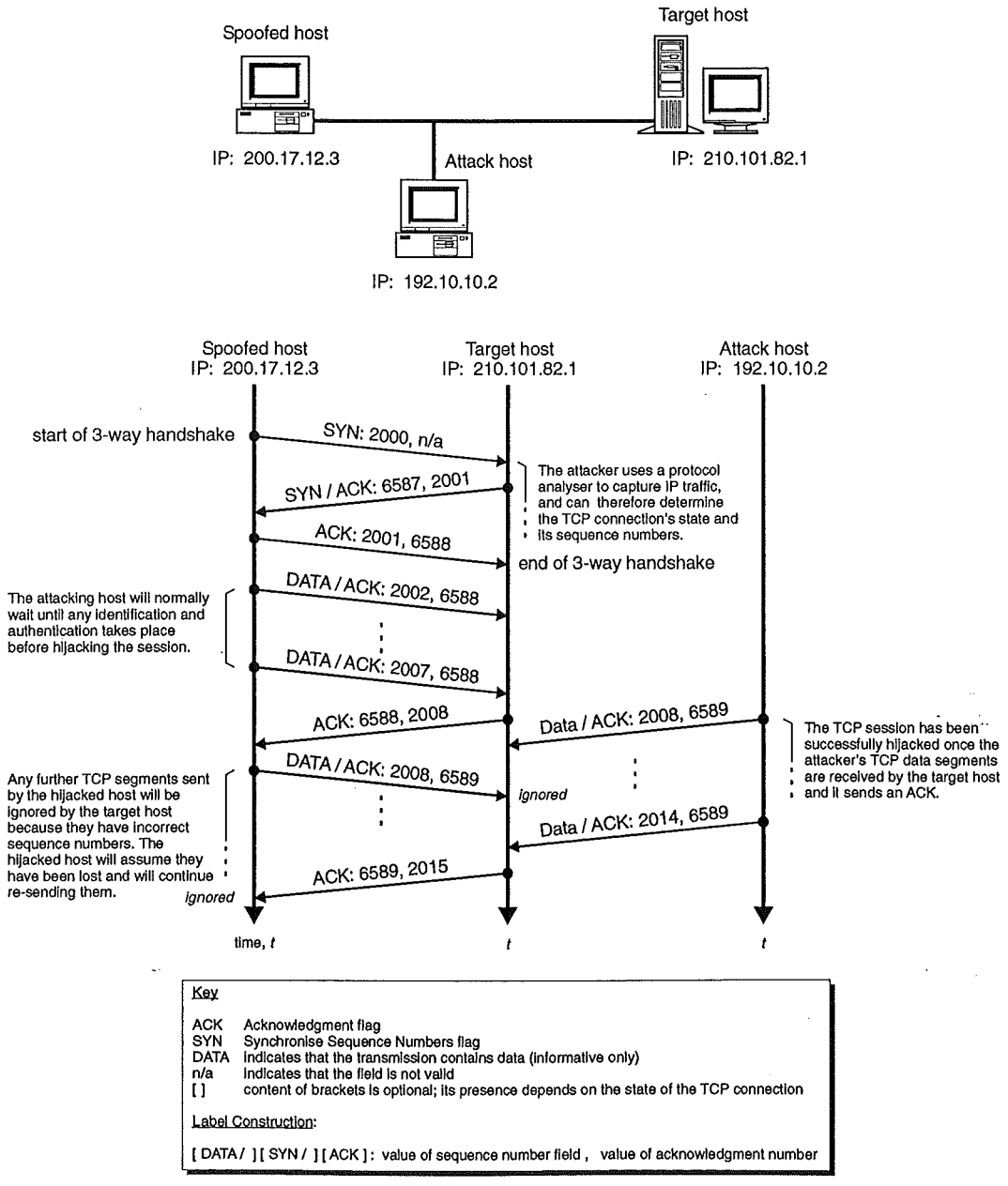

Figure 3-4 Example of a TCP session hijack.

Countermeasure

Again, the simplest and most effective defence against IP spoofing, TCP spoofing, and TCP session hijacking lies with those organisations providing access to the Internet. If all of these organisations were responsible enough to prevent IP datagrams with source addresses originating from outside their networks from reaching the Internet, the attacks described above could not be carried out.

Unfortunately, there are many organisations that provide unregulated Internet access. Therefore other means for protecting against spoofing and hijacking attacks must be used. The simplest and most effective is for an organisation to block all IP datagrams from the Internet that are source routed, or that have source addresses originating from the internal network. A properly configured firewall can be used to enforce such a policy.

Also, trust relationships (e.g. rhosts) between hosts communicating across the Internet should never be permitted, unless they are used in conjunction with strong authentication and cryptography[21] — they are simply too vulnerable! In fact, strong authentication and cryptography should be used with all TCP services (e.g. such as Telnet, FTP, etc.) where it is possible that an untrusted user could gain more than a very basic control over the operating system hosting the service. For example, an anonymous FTP server that provides read-only access to files can be adequately protected by the security mechanisms in existing operating systems, such as UNIX, and Windows NT. It is also important to assess the threat, for instance, it is unlikely that an attacker would go to the trouble of hijacking an anonymous FTP session! However, providing remote FTP access across the Internet to the superuser for uncontrolled read and write access has far greater implications. In such a case both strong authentication and cryptography are required, because the risk to the operating system by allowing such a connection would be too high.

It is essential to understand the possible threats and vulnerabilities introduced by connecting to untrusted networks so that the risks can be accurately assessed. It is not enough to consider the risks posed by applications (e.g. FTP, Telnet, WWW, etc.) alone, it is equally important to understand the risks posed by the network protocols themselves, such as the TCP, the IP, and the many others outside the scope of this thesis (e.g. IPX/SPX, NETBUEI, SNA, etc.)

3.2.3 RST and FIN Attack

Description

As mentioned previously TCP segments have control flags which indicate the status of a segment (see Table 2-3, page 17). There are two flags in particular, RST and FIN, which can be used for denial-of-service attacks. Under normal circumstances the RST flag is used to reset a connection, while the FIN flag indicates that no more data will be sent. As with TCP session hijacking, the only requirement for this attack to be practical is that the attacker must have access to the IP datagrams sent between the target and spoofed host. This is necessary so that a protocol analyser can be used to collect the IP datagrams and obtain the correct sequence number.

For a RST or FIN to be accepted, the TCP segment need only have the correct sequence number as the ANF is not used (i.e. there is no ACK in a RST segment). Therefore, the attacker simply analyses the IP datagrams in the connection between the target and spoofed host, and calculates (from the target host's ACKs) the sequence number that the target host would expect the next TCP segment from the spoofed host to contain. The attacker then generates a TCP segment with the RST flag set and sends it, in a spoofed IP datagram (i.e. containing the spoofed host's IP address in the source address field), to the target host. On receipt, the target host will close the connection with the spoofed host.

A very similar attack can be launched with the FIN flag, which is the normal way that a TCP connection is closed. The attacker uses a protocol analyser to predict the correct sequence number, using it to

[21] It is important to note that strong authentication and cryptography are not mutually exclusive. For example SSL can provide session encryption and strongly authenticate both the client and server (see Section 7.7).

construct a TCP segment with the FIN flag set. This is then sent to the target host which assumes that the spoofed host has no more data to send. Any further TCP segments sent by the spoofed host are ignored because the target host assumes they are network errors. The advantage of a FIN based attack is that the TCP mandates that on receiving a segment with the FIN flag set, the host must reply with one of its own. The beauty of this attack, from the attacker's perspective, is that they can be 100% guaranteed that their attack was successful!

Countermeasures

Normally, RST and FIN attacks are only applicable to the internal networks of an organisation. The reason for this is that an attacker needs to analyse the IP datagrams sent by either the target or spoofed host to determine the correct sequence number. For the attacks to be carried out on the Internet the attacker would have to have access to an Internet routing node at some point between the hosts being attacked — for most attackers access to such resources is impossible.

Denial-of-service attacks can prove to be particularly malicious. Take for example a critical online database that has an HTML interface which allows users to enter data. A malicious attacker could continually interrupt the commit phase (i.e. where the data is sent from the WWW-browser to the WWW-server) to prevent users from completing their work, or to corrupt the database. As a further example consider a WWW-server that provides information to users, an attacker could indiscriminately close connections during downloads causing many browsers to hang. These attacks would cause a great deal of confusion and be particularly difficult to resolve, i.e. is it a software, network, or hardware fault? Assuming the attacker does not wish to be caught they would stop their attack once an investigating was initiated and resume it once the investigation had finished — the infuriating, unpredictable, intermittent fault!

Unfortunately, configuring routers and gateways on the internal network to block such attacks is difficult, and often impracticable because of the distributed nature of user groups and information resources. In such environments there is little that can be done to protect against such denial-of-service attacks.

3.2.4 Ping O' Death

Description

The Ping program tests whether a host is reachable by sending it an ICMP echo request message and receiving an ICMP echo in reply. Ping also measures the round-trip time to the host, which provides an indication as to how distant the host is, and is helpful for determining whether the intervening network is congested.

IP datagrams (see Section 2.3) can be a maximum size of 65,535 (2^{16}-1) octets, which includes the header length (typically 20 octets if no IP options are specified). Datagrams that are larger than the maximum size that the underlying Link-layer can handle, known as the *Maximum Transmission Unit* (MTU), are fragmented into smaller datagrams which are then reassembled by the receiver. For Ethernet based networks the MTU is typically 1500 octets, while on the Internet the MTU is usually 576 octets.

The ICMP echo request resides within the IP datagram, and consists of eight octets of ICMP header information (RFC-792 [Postel, 1981c]) followed by the number of data octets in the "Ping" request. Hence the maximum allowable size of the data area is 65,535 - 20 - 8 = 65507 octets.

What makes the "Ping O' Death" attack possible is the ability to send an echo request datagram with more than 65507 octets of data, and because of the way IP fragmentation is performed. IP fragmentation relies on an offset value in each fragment to determine the order in which the individual fragments should be reassembled. Thus on the last fragment, it is possible to combine a valid offset with a suitable fragment size such that (offset + size) > 65535. Since operating systems typically do not process the datagram until they have reassembled all the fragments, there exists the possibility of

overflowing internal variables, and buffers which can lead to system crashes, reboots, kernel dumps, etc.

Unfortunately, "Ping O' Death" is easy to exploit, especially for those that have operating systems that allow users to send Pings of illegal size, such as Windows 95, Windows NT, and Linux. The following command is all that is needed to launch the attack from Windows 95:

```
> ping -l 65510 your.host.ip.address
```

Windows 95 will reply with "Request Timed Out", which means that the Ping was not answered, either because the remote host has correctly ignored the illegal Ping; or because it is now "dead" — it is that simple!

Countermeasure

Once it has been determined that hosts are at risk, the best solution is to obtain patches for the operating systems involved. Fortunately, the "Ping O' Death" attack is now mainly of historical interest as most operating systems released since early 1996 are immune[22], or have patches freely available. The attack is only possible because of insufficient error handling within the effected operating systems, not because of vulnerabilities inherent in the IP protocol itself.

However, if patches are not available a quick solution is to block Ping at the firewall. Unfortunately, blocking Ping messages also prevents legitimate use and may prevent certain applications from functioning properly. A better solution than blocking all Pings is to block only fragmented Pings. This allows common and legitimate 64-byte Pings through on most systems, while blocking those that are larger than the MTU.

Although the focus here is on Ping, it is important to consider that this attack is in theory applicable to any protocol that relies on IPv4 datagrams but cannot deal with those larger than 2^{16}-1 octets. Thus, it is possible that protocols such as TCP, UDP, and even IPX could be effected. The only completely effective solution is to secure the operating system against buffer overflows, and variables containing illegal values, when reconstructing IP fragments.

3.3 Threats to Standard TCP/IP Services

TCP/IP supports the operation of a number of well known services (i.e. applications). Traditionally each of these services have been associated with one or more vulnerabilities. Only applications that are commonly available on a number of operating systems, including UNIX, and Windows NT, are described here.

The intention is not to provide a detailed discussion about all applications that exist and have potentially exploitable vulnerabilities. Instead the following Sections are intended to provide an overview of the types of problems that are common to applications not included here, and to provide examples of the threats and vulnerabilities that those implementing Internet, Intranet, and Extranet networks should be aware of. For complete and detailed information about many other applications and their vulnerabilities the reader should consult [Cheswick et al., 1994] [Garfinkel et al., 1996] and [Hare et al., 1996].

[22] An unofficial WWW-site providing information, and a list of affected (including available patches) and unaffected operating systems is available at **http://www.sophist.demon.co.uk/ping/index.html**

3.3.1 Simple Mail Transport Protocol (SMTP)

Description

The *Simple Mail Transport Protocol* (SMTP), RFC 821 [Postel, 1982], is used as the basis for most email. Email is the most popular Internet service [Caceres et al., 1991], allowing people to communicate by exchanging electronic messages globally. These messages take anywhere from a few seconds to a couple of hours to be delivered. An added attraction is the relatively low cost of sending large messages. Combined, these benefits give users a convincing argument for access to email, and thus the connection of their systems to the Internet.

For a full and easy to read description of SMTP the reader is urged to consult [Stevens, 1994]. It must be noted that SMTP is a developing protocol, and as such, new threats could evolve. RFC 1425 [Klensin, 1993] defines the framework for adding extensions to SMTP.

Threats

SMTP used by itself is a fairly benign protocol, containing only eight basic commands. These are HELO, MAIL, RCPT, DATA, QUIT, VRFY, NOOP, and TURN. There are two security threats associated with these commands;

- Denial-of-Service
- Information gathering

Denial-of-service attacks based on SMTP are aimed at flooding a network or computer with large email messages to prevent legitimate use. In most cases a computer is affected because it cannot handle large messages e.g. > 1 Megabyte, or cannot handle the load created by receiving large numbers of messages at the same time, or running out of storage space.

For example the Computer Fraud and Security journal [CFS, 1996a] reported that a disgruntled university student was arrested for "mail bombing" the Monmouth University computer system in New Jersey. The attack caused massive disruption to the system for two days by generating 24,000 email messages, inundating the computers and paralysing the network. To get the systems functioning again required 44 hours of work, at an approximate cost of (US)$4,400.

The second more subtle attack involves information gathering designed to provide the hacker with useful information about a computer system and its users. For instance the VRFY command sometimes translates a user's mail alias into their login name. This can be used to identify the more promising accounts to attack, with tools such as Crack.

Most problems arise when SMTP is implemented as a large application, such as *sendmail* [Costales et al., 1993]. The threat comes from bugs, which inherently manifest themselves within large programs, and configuration problems such as giving the application higher privilege. These problems enabled one of the most famous Internet security incidents — the Internet Worm [Spafford, 1989] to take place.

Other problems also exist with email attachments, and automated execution of encoded messages such as Multipurpose Internet Mail Extensions (MIME). MIME allows specific actions to be encoded in email messages. These actions can request files to be automatically retrieved and returned to the message initiator.

MIME can also be used to transfer executable programs and Postscript files, which can themselves perform dangerous actions. These existing security threats are very applicable to new, network oriented, programming paradigms such as Java and ActiveX (see Section 3.3.7).

3.3.2 Telnet

Description

Telnet, RFC 854 [Postel, 1983], is designed to enable communication between any host, regardless of the operating system. Telnet provides simple character based terminal access, and usually requires the user to login with an account name and password.

Threats

The biggest threat comes during login when initiating the Telnet session, as standard Telnet does not protect the transmission of the user's account name or password. Anyone monitoring the Telnet login IP datagrams over the network can capture this information.

As with any protocol each step is predictable, therefore a packet sniffer can be configured to simply detect any Telnet session and record the IP datagrams containing the account name and password.

Other threats exist, for example the Telnet program itself could have been compromised to record passwords and account names. A description of such a case is available in [Safford et al., 1993a].

To protect against sniffing attacks a number of secure versions of Telnet have been implemented [Borman, 1993] [Safford et al., 1993b]. These versions of Telnet usually encrypt both the password and session contents which prevents an attacker from obtaining any useful information.

3.3.3 Network Time Protocol (NTP)

Description

The *Network Time Protocol* (NTP), RFC 1305 [Mills, 1992], is used to synchronise the clocks of hosts connected to the Internet. The correct time is generated by extremely accurate atomic clocks which provide national time synchronisation. Time updates are propagated through a directed hierarchy of Internet hosts. The propagation path must not contain any loops as this would cause erroneous time transfers.

NTP provides accuracy of 10ms or better; with such accuracy comes the ability to match log files from different systems. This has proved beneficial when matching audit logs from different systems and allows an attacker's actions to be replayed. It also provides a mechanism for cryptographic protocols to generate timestamps for authentication purposes.

Threats

Attacks on NTP focus on altering a target's sense of time. If this succeeds, a time based authentication protocol can be subverted by replaying a previous successful authentication sequence. Protection against these attacks is provided in newer versions of NTP which provide cryptographic message authentication. NTP specifies that authentication be carried out on a hop-by-hop basis. It is therefore possible for an attacker to subvert a system on which the target's NTP daemon relies, and thus subvert the target system as previously described. To ensure protection against this type of attack all sources of NTP information authenticate their sources, and so on back to the root NTP server.

3.3.4 Finger and Whois

Description

The Finger protocol, RFC 742 [Harrenstien, 1977], provides information on users of a specific host. Generally it is used to find out the account name of a user and/or whether they are logged on. In most cases the person using this command has no more sinister motives than sending mail.

The Whois protocol, RFC 812 [Harrenstien, 1982], provides contact information such as account name, telephone number and address. It is useful for looking up people on systems when you do not have there full name. For example typing "whois smith" will return a list of people with "smith" in there name.

Threats

Finger can be used by hackers to collect useful information, such as account names, and compile login profiles i.e. the best time to attack the system is when the system administrator has finished for the day, or better still, on vacation. Other useful information supplied can be the date a user last logged in, and a user's ".plan" file which often contains useful personal information. This information can be used to identify promising targets, and provide contextual information to attackers for use with tools such as Crack. The following extract is from RFC 742 and expresses the philosophical nature of Finger. It reflects well the openness of early networks and contrasts starkly with the more security conscious 1990's.

> *"To fulfil the basic intent of the Name/Finger programs, the returned list should include at least the full names of each user and the physical locations of their terminals insofar as they can be determined. Including the job name and idle time (number of minutes since last typein, or since last job activity) is also reasonable and useful."*

The "Finger Bomb" is an interesting use of Finger to launch denial-of-service attacks against systems (Note: this attack has been patched on newer Finger services). Some Finger services allow the redirection of Finger to remote sites. To Finger through several sites, an intruder could use:

> finger username@hostA@hostB

The Finger will go through host B then to host A. This helps attackers to remain anonymous because host A will see a Finger coming from host B instead of the original host. This technique has also been used to go through firewalls that have not been properly configured. This can happen by using the command:

> finger user@host@firewall

On vulnerable hosts a denial-of-service attack can be launched by typing:

> finger username@@@@@@@@@@@@@@@@@@@@@hostA

The repeated @ causes the Finger to recursively Finger the same machine repeatedly till the memory and hard drive swap space fill up. This causes the machine to crash or slow to an unusable speed.

The best countermeasure available to address the threat from Finger is to disable it entirely. If this is not possible then Finger should only be allowed to retrieve user information from a sanitised database.

The Whois protocol is susceptible to the same types of abuse as the Finger protocol, however it does not reveal detailed information about users access habits.

3.3.5 Network File System (NFS)

Description

The *Network File System* (NFS), RFC 1094 [Sun Microsystems, 1989], protocol provides transparent remote access to shared files across networks, and is designed to be portable across different machines, operating systems, network architectures, and transport protocols. This portability is achieved through the use of Remote Procedure Call's (RPC) [Sun Microsystems, 1988].

To ensure robust NFS access in the event of system reboots and device failures (e.g. bridges and routers) the NFS server is stateless, unlike the clients which retain state. When an NFS server becomes unreachable its clients continue to send requests until they receive a reply. Thus, the client's functioning is not adversely effected by the loss of an NFS server.

Threats

All files and directories on an NFS server are identified by unique strings known as file handles. A threat is introduced if a client program obtains and retains a root file handle at mount time, which is usually when the NFS server is re-booted. This is possible due to the inadequacies NFS access controls.

Once access to the file system has been achieved it is possible to change file access controls, and create subversive programs and place them in search paths so that the real ones are not used e.g. trapdoor or password gathering programs.

3.3.6 File Transfer Protocol (FTP)

Description

The FTP, RFC 959 [Postel, 1985], enables the transfer of character and binary files across a network. The design philosophy does not dictate a specific host, operating system or file structure — it is completely independent.

An FTP server uses two TCP ports to transfer a file. Control Connection is established on Port 21, and Data Connection on Port 20. The FTP client is free to choose any available port.

FTP has become the standard for publishing software, data, and documents on the Internet. However newer protocols such as *Hyper Text Transfer Protocol* (HTTP) using the *Hyper Text Markup Language* (HTML) are becoming popular for documents.

Threats

The major threat to FTP comes from improperly managed FTP services. For example if an organisation runs a public FTP service but does not separate its sensitive organisational data, then with today's network speeds it may be possible to download all the sensitive data in a matter of minutes. FTP services should be restricted to certain, well managed, file areas.

FTP has been used to gain access to password and remote host files by exploiting deficiencies in management of the service. For example, if file areas are not controlled then the user is able to change access controls to files. It may be possible to insert false password or remote host files, which can then be used to gain access to other hosts.

Like Telnet, the standard FTP protocol does not encrypt passwords that are required for the user to login to a system, so there is a high risk that the password can be compromised by anyone listening into the network. FTP sites are also used as promulgation points for pirated software.

3.3.7 World Wide Web (WWW)

Description

The WWW is made up of a collection of protocols specifically designed for exchanging information over the Internet. The original WWW protocols included *Gopher*, *Wide Area Information Servers* (WAIS), and *Archie*, however, the past four years have seen the introduction of the HTTP that has revolutionised the Internet. In fact, most laypersons associate the term WWW exclusively with HTTP.

These protocols are generally used by clients to query servers for specific files. HTTP also implements the client/server model of document retrieval, in this case the client, called a "browser", is usually capable of multimedia support. The server, referred to as a WWW-server, functions in a similar manner to a standard file server, simply sending the requested documents to the browser. However, WWW-servers are also capable of running programs to create HTML documents dynamically as they are requested, this makes them very useful for maintaining documents in dynamic environments. In fact HTTP was originally developed by physicists at CERN laboratories as a means of exchanging papers pertaining to their research. These documents where constructed using HTML which is based on the *Standard Generalised Markup Language* (SGML).

What makes HTML so attractive is that a document can incorporate small programs that allow the content to become dynamic. These programs, referred to as *executable content*, can either be included as scripts within the document (e.g. Java Script), or as compiled programs which are loaded separately when the document is accessed (e.g. Java and ActiveX).

Threats

Sometimes files are returned which contain format tags, these are used to identify the program necessary to view, or execute the files. This problem is similar to trusting MIME encoded email messages. In fact, this is a major problem with HTML documents containing executable content.

Most organisations that are connected to the Internet provide their employees with browsers so they can access WWW-servers. Unfortunately, to achieve this, firewalls and network guards must be configured to permit outgoing HTTP connections. This means that unknown programs contained in HTML pages can be downloaded onto a user's computer and executed, effectively bypassing the firewall and any security policy that attempts to control the unauthorised use of untrusted software!

Fortunately, several solutions have emerged to deal with this problem. The first limits the access that the software has to system resources. For example, Java Script runs within the environment created by the browser and does not have direct access to system resources (e.g. hard disk, device drivers, memory, etc). Similar constraints are also applied to Java applets, although these can be relaxed to some extent by the user. The most dangerous executable content is Microsoft's ActiveX, these programs, known as "controls", are in fact executable binaries (i.e. compiled Microsoft Windows C++ programs). They are executed by the browser in the same manner that a user runs a program, because of this the ActiveX control has the same access rights to system resources as the user running the browser. For example, an ActiveX control downloaded by a user with administrator privileges would have full control of the computer, and possibly other machines if connected to a network.

To address the problem users have in deciding whether executable content can be "trusted", Netscape and Microsoft have developed technologies based on public-key cryptography allow Java applet and ActiveX control code to be digitally signed. A browser that downloads a signed ActiveX control or Java applet can check the signature against a list of trusted certificates, if signed correctly the user can choose to execute the program with confidence that it came from a trusted source. Browsers from Netscape and Microsoft are pre-loaded with certificates from a number of well respected organisations. The benefit of this technique is that an organisation can remove all default certificates and install their own, effectively restricting executable content to that developed by the organisation. This can be enforced because the above browsers can be configured to enforce particular security policies.

Another solution provided by many of the newer firewalls allows HTML tags (i.e. hyper-links) that load the executable content to be disabled. Some firewalls can also be configured to check the signatures of Java applets and ActiveX controls, and allow through only those signed by trusted certificates. This has the added benefit of enforcing the security policy at a central point, other than delegating it to the browser where it may be possible for a user to alter the security policy locally.

In most cases the solution to the problem of executable content is similar to FTP and other services. That is, services should be run in an enclosed environment with only enough privilege to perform their task.

3.3.8 X-Window System

Description

The X-Window system [Scheifler et al., 1992] is a client/server application which enables multiple clients to use the bit-mapped display managed by a server, which also manages the keyboard, and mouse. The client is an application program which runs on a host with the server or on a different host.

X-Windows requires a reliable, bi-directional stream protocol such as TCP. Communication between client and server consist of 8-bit bytes. On UNIX systems where the server and client are on the same host, UNIX domain protocols are used to reduce the overhead of the TCP protocol.

Threats

An application which connects to an X server is able to do a multitude of things, e.g. read the keyboard, print the screen, read mouse movements/button presses, simulate key-presses, resize windows etc. If an attacker can connect to a server and read the keyboard, the user will be compromised. It is possible for an attacker on the Internet to probe for X servers, as X server ports are assigned as $6000 + n$, where n is some small integer, usually 0.

The X-Windows system uses host based authentication. The server takes the network source address of the connecting application and compares it with a list of allowable sources. However, there is no protection from an attacker connecting from a trusted host.

Another protection mechanism makes use of a magic cookie; this is a secret byte string which the server and application share. Processes cannot connect to a server unless they contain this string. The problem is communicating the secret string between application and server over a generally unsecured network.

A similar cryptographic challenge/response protection mechanism exists, but suffers from the same key distribution problems as the magic cookie.

3.4 Summary

The Internet can be a dangerous place for those who are not prepared. This warning is supported by the 1998 joint CSI and FBI survey of computer crime. Although the Internet is perceived by many to pose the greatest threat to an organisations networks, the threat from dial-in connections and especially employees is just as great. Reality is that computer crime is costing organisations a great deal of money, though figures for New Zealand organisations are not available 241 organisations in the US reported a combined financial loss during 1997/98 of nearly US$235 billion dollars!

Much of this figure can be attributed to the attacks discussed here. Although it is possible that having appropriate countermeasures in place would have reduced this figure. Unfortunately, computer systems are only as secure as those that use them can be trusted. This is borne out by the figures from Table 3-2 in categories such as, unauthorised insider access, insider abuse of network, sabotage of data or networks, etc.

The TCP/IP suite was never intended to offer comprehensive, scaleable security mechanisms, and it is the lack of such mechanisms that underlie most of the problems with IPv4 and TCP. However, many solutions have been presented here and most are readily available without great expense. For example, there is little expense in ensuring that trust relationships (e.g. rlogin) do not exist, or in applying patches (e.g. Ping) and keeping them up-to-date.

Perhaps the most important point is that all organisations should act responsibly to prevent malicious traffic from reaching the Internet. As discussed in Section 3.2, most attacks to the IP and TCP (e.g. SYN flooding, IP spoofing, etc.) could be averted by preventing IP datagrams leaving an organisation's network if its source address did not originate from within. Unfortunately, not all organisations are so responsible, thus attacks which could be easily prevented are still possible.

It has also been shown that many applications pose significant risks to organisations. Most problems are caused through deficiencies in the implementation (e.g. buffer overflows, unhandled exceptions, etc.) Therefore, it is essential that applications are kept up-to-date by applying patches or service packs that address new exploitable vulnerabilities. Other problems are caused by uneducated users or shortcomings in the organisations security policy. For example, Table 3-2 estimates that for the 1997/98 period US$20 million dollars was lost to virus incidents. Also, it remains to be seen what problems, and financial losses, new WWW technologies (e.g. ActiveX controls, Java applets) will inflict.

It is expected that IPSec (see Section 7.6) will solve many of the problems associated with existing TCP and IP implementations. However, deficiencies and errors in the implementation of applications, along with corrupt employees, will continue to introduce new generations of threats and vulnerabilities.

Chapter 4. Firewall Technology

4.1 Introduction

A firewall is a combination of components, or a system that functions to enforce an access policy between two networks. The combination of components are known as a *firewall architecture*, which is directly responsible for protecting a single connection route between internal networks, or between internal and external networks. An *internal network* is defined as one that the organisation has control over, and is therefore considered a *trusted* network. In contrast, an *external network* is defined as one that the organisation has no control over, and is therefore considered an *untrusted* network. This thesis considers firewalls in the context of connecting internal networks to external networks (e.g. the Internet).

However, firewall architectures are equally applicable to any two (or more) networks that have different security policies, or do not share common level of trust. For example, if an organisation has a LAN used by the Accounts Department and wishes to protect it from other departmental LAN's. A firewall can be placed between them to control the types of access permitted to employees in the different departments. Thus, all aspects of this thesis are equally applicable to the protection of internal networks from one another.

A firewall architecture possesses the following properties [NCSA, 1996]:

- all traffic from the internal to external, and vice-versa, must pass through it,
- only authorised traffic, as defined in relevant security policy, is allowed to pass through it, and
- the firewall architecture itself is immune from penetration

Implementing a secure firewall architecture is dependant on the amount of resources the organisation is willing to expend, and the level of risk the organisation is willing to accept.

4.2 Firewall Terminology

To provide a common basis for the discussion of firewall architectures, the following terminology is introduced:

Screening-Router – A basic component of most firewall architectures. A screening-router is usually a commercial router, although it can be a host with packet filtering capabilities. Typically screening-routers have the ability to control the flow of traffic between specific networks or hosts, at the IP layer. A screening-router can be the sole component of a firewall architecture.

Bastion-Host – A bastion-host is analogous to a highly fortified castle; providing a central point for the protection of the surrounding countryside. Therefore the bastion-host is identified as a critical point in the security of a network. Bastion-hosts have extra attention paid to them which may be in the form of regular audits, have less or altered software/hardware, etc., to improve security. Bastion-hosts are commonly used to implement Application-level gateways (see below).

Gateway – The terms "gateway" and "security gateway" are used in firewall literature to promote the idea of firewall as a single point which controls all communication between two or more networks. These terms are not preferred because they suggest that the firewall exists as a single component in the network, and possibly performs processing at all seven layers of the

OSI model — which is generally not the case. The preferred term is "architecture" or "firewall architecture", which promotes the idea of various components acting synergistically to protect a network from external threats. However, the term "gateway" is retained to provide consistency with existing literature.

Dual-Homed Gateway – A dual-homed gateway (see Section 5.3) is implemented without a screening-router. The dual-homed gateway consists of two network interfaces, one connected to the external network and the other to the internal network. An important requirement is that TCP/IP forwarding is disabled to ensure that all traffic between the external and internal networks is inspected by the bastion-host. Thus, direct communication between networks is prohibited. By definition a dual-homed gateway is a bastion-host.

Screened-Host Gateway – The screened-host gateway (see Section 5.4) is possibly the most common firewall architecture. It is implemented using a screening-router and a bastion-host. In most cases the bastion-host is located on the internal network. However, the screening-router is configured so only the bastion-host is visible from the external network. The screening-router can be used to limit the number of reachable services by blocking traffic based on port number.

Screened-Subnet – A screened-subnet is a firewall architecture in which an isolated network segment is created between two screening-routers. The bastion-host and other sacrificial hosts, such as WWW-servers, are placed on the isolated network segment. The screening-routers are configured to permit traffic from the external network to reach hosts on the subnet only, external traffic attempting to connect directly to the internal network is blocked. The bastion-host is only required if Application-level firewall architectures are to be supported.

Proxy – A proxy is an application program which acts on behalf of another application, such as a WWW-server. For example a WWW-proxy passes *Uniform Resource Locator*[23] (URL) requests/responses between browsers on the external network and the WWW-server on the internal network, or vice-versa. The same principles apply to other types of proxy. Importantly a proxy understands the protocol it is representing which enables the proxy to alter and monitor (meaningfully) the traffic it exchanges. The benefit of this is that the proxy can alter the protocol, for example, to support advanced authentication, or to provide improved auditing and logging capabilities.

Proxies are not always security related. For example, proxies have been developed to provide a caching service for URLs. When the proxy receives a request for a URL, it looks to see whether it is already located in its local cache. If found, it returns the document immediately, otherwise it is fetched from the remote server with a copy being saved in the local cache. Such a proxy is often termed a "proxy-server". The proxy-server usually incorporates a mechanism to expire cached documents according to their age, size, and access history.

In most cases proxies are transparent to the connecting hosts which simply appear to connect to the expected service unimpeded. The most significant problem with proxies is that they have to be written specifically to mimic the application they represent (i.e. essentially they must implement any communication protocols provided by the original application). Proxies form the basis of Application-layer firewalls.

IP masquerading – IP masquerading, or *address translation*, is a technique used by firewalls and security gateways to hide the use of unofficially registered IP addresses, and the topology of internal networks. Address translation can be achieved at the Network-layer by the operating system, or at the Application-layer using a proxy. Essentially, in both cases the IP source address of each datagram is replaced with an official address (usually that of the firewall's external NIC) before being sent on to the external network.

[23] A Uniform Resource Locator (URL) is used to specify the location of an object on the Internet, such as a file or a newsgroup. URLs are used extensively on the WWW, and within HTML documents to specify the target of a hyperlink which is often another HTML document (possibly stored on another computer). For example, the Internet-Draft for the URL specification can be obtained by following the URL; **http://www.w3.org/hypertext/WWW/Addressing/Addressing.html**

Wrapper – Wrappers are a recent addition to UNIX security. They literally wrap around a program to enforce a higher degree of security than the wrapped program could achieve on its own. A common use for wrappers is to control the amount of information reaching the wrapped program, e.g. potentially dangerous commands can be filtered out. They can also provide extra security functionality, such as authentication, auditing and logging. Wrappers have a number of benefits, for instance, they are generally small programs which are easier to validate. In addition, wrapped programs can be upgraded without the need to rewrite the wrapper. Of course, any bugs that exist in the wrapped program are still potential vulnerabilities.

Application-Server – An application-server provide a specific service, such as SMTP, FTP, WWW, or DNS. Application-servers usually operate at the Session, Presentation and Application-level of the OSI model. Application-servers are prone to security vulnerabilities, most often because of their size and complexity. Perhaps the most famous example is the sendmail hole that was exploited by the Internet worm (see Section 3.3.1).

If application-servers are run on a firewall then any exploitable vulnerabilities could be used to directly compromise the firewall. It is commonly accepted that application-servers should not be run on a firewall, rather proxies or at least wrappers should be used.

Circuit-Level Gateway – Circuit-level gateways are similar to application-level gateways with a single distinction. Instead of connections being mediated by the firewall, a virtual circuit is created between the external and internal hosts. This results in creating a hole in the firewall. Generally, circuit-level gateways are used to relay TCP connections from internal to external hosts. A certain amount of trust must be placed in the host opening the circuit, as the level of control the firewall has over the connection once it is established is less than that of the application-level gateway.

Hybrid Gateways – Usually the term "hybrid" is used with firewalls or gateways that are composed of non-standard network components. Hybrid gateways may also use protocols other than TCP/IP, such as Novell's IPX/SPX network protocol. Protection of Hybrid gateways may rely on routers, or proprietary mechanisms specifically designed for the hardware or software being used. In general the concepts discussed in this thesis are equally applicable to hybrid firewalls or gateways.

4.3 The OSI model

The OSI model (see Table 4-1, page 46, for an overview) is used to relate the various firewall components to their functionality. The OSI model provides a clearer reference than the TCP/IP model. Figure 4-1, page 46, presents a comparison between the OSI an TCP/IP communication architectures. A detailed description of the OSI model can be found in [Stallings, 1991].

Firewall architectures consist of components at two levels; the packet-level, and application-level. The packet-level components perform their actions based on information contained within the IP and TCP headers. In relation to the OSI model, the IP and TCP layers correspond to the network and transport levels respectively. The firewall architecture commonly used at this level is the screening-router.

Table 4-1 An overview of each layers of the OSI model.

Layer	Name	Description
1	Physical	Concerned with the transmission of an unstructured bit stream over a physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium.
2	Data link	Provides for the reliable transfer of information across the physical link; sends frames of data with the necessary synchronisation, error control, and flow control.
3	Network	Provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections.
4	Transport	Provides reliable transparent transfer of data between end points; provides end-to-end recovery and flow control.
5	Session	Provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between communicating applications.
6	Presentation	Provides independence to the application process from differences in data representation (syntax).
7	Application	Provides a means for application processes to access the OSI environment, and provides distributed information services.

Application-level components generally perform their actions based on the protocols of the services being used. With respect to the OSI model, application-level firewalls relate to the application, presentation, and session levels of the OSI model. A bastion-host is generally used to implement the application-level component (see Figure 4-2, page 47). As Application-level components are at the highest level of the OSI stack and TCP/IP suite, it is possible for them to make use of information from lower levels. For instance an application-server could use IP header information to check the source address of the client connecting to it.

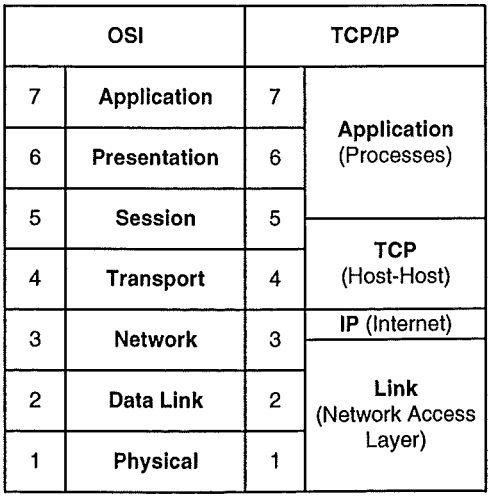

Figure 4-1 Comparison of OSI and TCP/IP communications architectures.

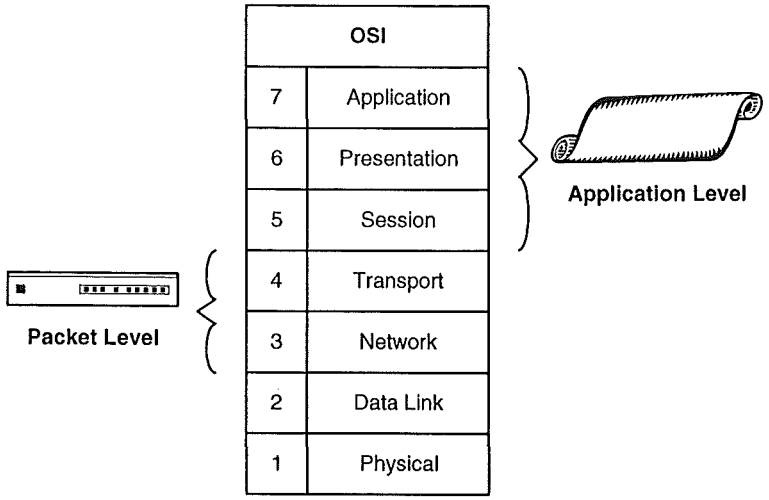

Figure 4-2 OSI model in relation to the various firewall architectures.

4.4 Defining Boundaries

It is important when considering firewall architectures to understand which parts of the organisation are at risk from the external network. The organisations network boundary is known as the *security perimeter* [Hare et al., 1996]. The responsibility of the firewall architecture is to protect this boundary from unauthorised transgression.

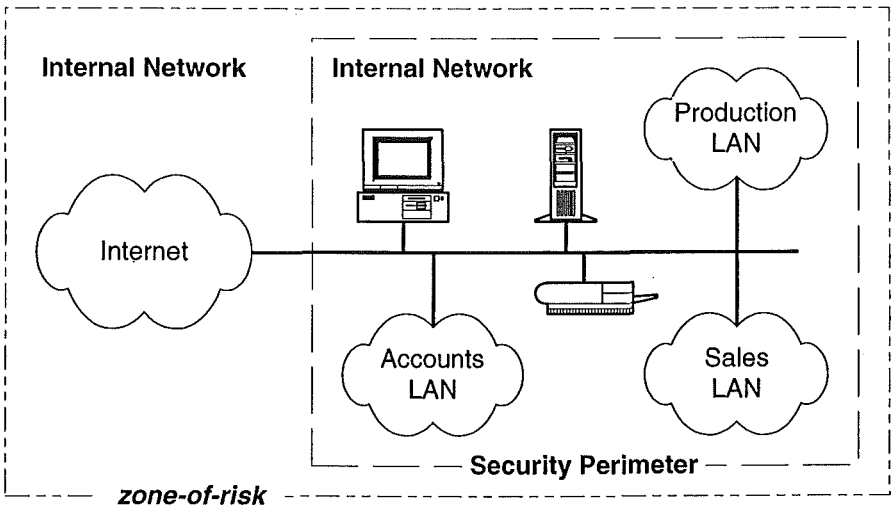

Figure 4-3 The zone-of-risk for an organisational network connected to the Internet without a firewall architecture.

It is also helpful when considering network security to define a *zone-of-risk*. The zone-of-risk is a measure of the number of internal network components, such as hosts or routers, which are accessible from external networks. Initially the zone-of-risk includes all networks directly accessible (i.e. there are no security measures in place, such as firewalls) through external networks, such as the Internet. This situation is presented in Figure 4-3. Generally, the zone-of-risk is restricted to TCP/IP capable networks. However, networks and hosts using other protocols may also be vulnerable, as attackers can

take advantage of the common protocols that exist to communicate between different network architectures.

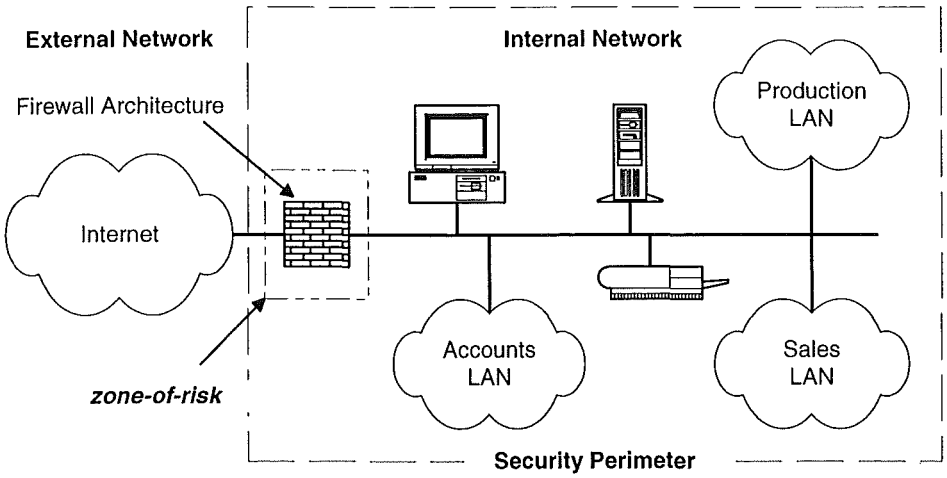

Figure 4-4 The zone-of-risk with firewall architecture in place.

The aim of a firewall architecture is to minimise an organisation's zone-of-risk by removing the number of network components which can be directly attacked from the external network. In other words the firewall architecture *becomes* the zone-of-risk for the entire organisational network (see Figure 4-4).

4.5 The Role of a Security Policy

A security policy is a prerequisite for any computer system, and should be promulgated to all members of an organisation. This ensures that all users understand their responsibilities, and rights in regards to the security of the organisations information systems. The security policy for a computer system should be concise and unambiguous, while providing the basis for the rules, regulations and procedures required for a detailed information systems security policy. A firewall security policy is the most important aspect of a firewall architecture, its specifications should be used to determine the design and performance requirements of the firewall [Menkus, 1995].

The firewall should implement the security policy defined by an organisations senior executives. Its purpose is to ensure the security of the organisations information systems from internal or external threats. The level of protection provided should be balanced in relation to the perceived threats.

The security policy sets the acceptable limits of behaviour on a system and is therefore fundamental to the operation of a firewall. What constitutes acceptable behaviour is defined by the underlying organisational philosophy towards information systems security.

The two philosophies which exist when considering a firewall security policy:

"That which is not expressly permitted is prohibited."

This is the most secure and attempts to mandate what can traverse the network. The second resides at the other end of the spectrum and reflects a totally permissive policy;

"That which is not expressly prohibited is permitted."

The later philosophy can be particularly difficult to define in a security policy. It requires all possible services to be identified and a decision to be made on the availability of each to the end-user.

Incorporating and maintaining this philosophy is a near impossible task for a systems administrator, due to the dynamic nature of computer systems, and the continually expanding requirements of end-users. The greatest problems arise when a security policy based on this philosophy neglects to prohibit a service that is potentially dangerous. If someone then exploits this service the organisation has no one to blame but itself.

It is much simpler, and safer, to implement the former philosophy that mandates which services will be made available to users. Using this approach services can be provided in response to user needs, and are under the direct control of the systems administrator.

The *National Computer Security Association* (now the *International Computer Security Association*) has released a firewall policy guide [NCSA, 1996] intended to promote a better understanding of firewalls among executives, information managers, system administrators, and MIS staff. The guide introduces two levels of network policy which are fundamental to the design, installation and use of a firewall architecture:

- *Network Service Access Policy* (NSAP) — a higher-level, issue specific policy which defines those services (e.g. Telnet, FTP, NNTP etc.) that will be allowed or explicitly denied from the internal network, plus the way in which these services will be allowed or explicitly denied from the internal network, plus the way in which these services will be used, and the conditions for exceptions to this policy.

- *Firewall Design Policy* (FDP) — a lower-level policy which describes how the firewall architecture will actually go about restricting the access to, and filtering of the services as defined in the NSAP.

4.5.1 Network Service Access Policy (NSAP)

The NSAP defines which services are to be explicitly allowed or denied between trusted and untrusted networks, together with the way in which these services are to be used and any conditions for exception to this policy.

The NSAP should be an extension to existing business policy which will have already addressed the following issues:

- *Information Value* – what value does management place on its information resources?

- *Responsibility* – who is responsible for ensuring the protection of the organisations information from untrusted networks?

- *Commitment* – what is the organisation's commitment to protecting its information resources?

- *Domains* – what domains should or should not be protected?

Further business policy should already have implemented controls on such systems as:

- virus scanning

- physical security access

- floppy disk controls

- RAID back-up systems

At the highest level the organisational policy might state:

- information is the strategic resource for the organisation.
- the availability, integrity, authenticity and confidentiality of the information will be protected by every cost-effective measure possible.
- ensuring the availability, integrity, authenticity and confidentiality of the information is a priority for all users at all levels of the company.

Below this level specific policies are implemented which cover issues such as:

- access to services (dial-in, dial-out)
- version controls
- user authentication
- trusted/untrusted network access

It is at this level that the firewall's NSAP is formulated.

The NSAP must be drafted before the firewall is implemented. It must provide a balance between protecting the trusted network from known risks while providing users with convenient access to the untrusted network. Further, if a firewall denies access to certain services on an untrusted network, it is essential that the NSAP ensures that these controls are not circumvented or disabled. A typical NSAP might:

- allow no access to applications or services on the trusted network from the Internet.
- as above but allow access to a subset of applications or services by way of a secure server (e.g. bastion-host).
- allow access from the Internet to selected applications on the trusted network (e.g. email) in conjunction with strict authentication procedures (e.g. challenge/response and one time password controls).

4.5.2 Firewall Design Policy (FDP)

FDP defines how the firewall implements restricted access and service filtering specified by the NSAP and addresses issues such as:

- IP address filtering
- encryption tunnelling
- secure socket control to facilitate application access
- audit and accounting control

This policy is specific to the firewall and defines the rules and procedures necessary to implement the NSAP, but must take account of the capabilities and limitations of the particular firewall platform as well as the threats and vulnerabilities associated with TCP/IP. For example, if the NSAP forbids access to all applications on the trusted network, then implementing a firewall by way of a packet filtering router is extremely risky.

In principle a firewall can:

- permit any service unless it is specifically disallowed
- deny any service unless it is specifically permitted

However, as stated previously only the latter option is practical. The first option might unintentionally allow denied services to run on non-standard TCP/UDP ports. Further, some services such as FTP, RPC and X-Windows are difficult to filter [Cheswick et al., 1994].

Depending upon the various security and flexibility requirements, some firewalls are more appropriate than others which means that the NSAP must be carefully designed before the firewall is implemented. For example dual-homed gateways (Section 5.3) and screened-subnets (Section 5.5) can both be used to implement a "deny all" firewall. However, the dual-homed gateway is cheaper but also less flexible than the screened-subnet.

In order to arrive at a successful design policy together with a platform which implements this policy, it is usual to start by restricting all access from the untrusted to the trusted network, and then to specify the following [NCSA, 1996].

- what Internet services will the organisation use (e.g. email, Telnet, FTP, WWW?)
- where will these services be used from (e.g. Intra-company, between branches, on a mobile or dial-in basis, by subsidiary organisations etc?)
- what additional security features will be needed (e.g. one-time password control, authentication procedures, encryption tunnels, secure sockets, point-to-point encryption, dial-in/dial-back procedures etc?)
- what risks result from the provision of these services (e.g. is 40-bit RSA cryptography adequate for certain Government or banking applications? Is dial-in access without strong authentication an acceptable risk?)
- what is the cost (financial, inconvenience) of providing these services? For example, how is key distribution handled? What is the cost of managing a dedicated authentication server?
- what is the balance between usability and security (e.g. if a particular service is too expensive or risky to use should its use be forbidden — thus creating great inconvenience?

Some services which are inherently insecure may, with the addition of certain technologies, be secured to pose little or no risk. For example a remote Telnet session can be extremely vulnerable to packet sniffing for passwords, and would pose a high risk when connecting a machine to a trusted network over an untrusted network such as the Internet. However, with the addition of encryption or strong authentication techniques this risk can be dramatically reduced.

Implementation of the firewall based upon these considerations requires careful use of risk analysis so that the calculated level of risk can be compared with that deemed to be acceptable according to overall company policy [White et al., 1996]. This may result in a change to the initial policy. For example, if the original NSAP denied all dial-in access, certain exceptions to this rule may need to be considered so as to allow for mobile users.

4.5.3 Sample Policies

Remote Access Policy

As a specific example a 'Remote User Advanced Authentication Policy' might address dial-in user access from the Internet as well as authorised users on travel or working from home. All such connections should use the strong authentication service of the firewall to access systems at the site. Policy should reflect that remote users may not access systems through unauthorised modems placed behind the firewall as it takes only one captured password or one uncontrolled modem line to enable a backdoor around the firewall.

Authorised users may also wish to have a dial-out capability to access those systems that cannot be reached through the Internet. These users need to recognise the vulnerabilities they may be creating if they are careless with modem access. A dial-out capability may easily become a dial-in capability if proper precautions are not taken.

Therefore, both dial-in and dial-out capabilities should be incorporated into the design of the firewall. Mandating outside users to go through strong authentication at the firewall should be forcefully reflected in policy. Policy might also prohibit the use of unauthorised modems attached to host systems and PCs on the organisations trusted network if the modem capability is offered through the firewall.

Since users could run *Serial Line IP* (SLIP) and *Point-to-Point Protocol* (PPP) to create new network connections into a site protected by a firewall they need to be considered as part of the overall access policy. Such connections are potentially a backdoor around even the best firewall architecture.

Information Server Policy

A site providing public access to an information server may wish to incorporate this access into the firewall design. While the information server itself creates specific security concerns, the information server need not become a vulnerability to the security of the protected site. Policy should reflect the idea that the security of the site will not be compromised in the provisioning of an information service. For example, a WWW-server that is intended to provide access for Internet users may not need to be behind the firewall at all as the information provided by the WWW-server resides on that machine, rather than being drawn from systems on the internal network. As long as the machine is regularly backed up it can operate unencumbered by a firewall and simply be restored if it is attacked.

It is useful to make a distinction between two fundamentally different types of traffic:

- information-server traffic (traffic concerned with retrieving information from an organisation's information server)
- business traffic such as email, file transfer, transaction services etc.

The two types of traffic have their own risks and do not necessarily need to be mixed with each other. Screened-subnet firewalls (Section 5.5) allow information servers to be located on a subnet and therefore to be isolated from other site systems. This reduces the chance that an information server could be compromised and then used to attack site systems.

4.5.4 Policy Evolution

Two considerations drive the formation of a FDP with respect to Internet connections:

- the risk to the organisation's internal information and systems from external threats, e.g. denial-of-service attacks, IP spoofing etc.

- the risk of sensitive organisational information being disclosed as it is transmitted across the Internet, e.g. password file capture, information leakage attacks (e.g. Finger) etc.

Once the FDP has been drafted, maintenance and review are important ongoing activities.

Maintenance of the FDP

Unlike many organisational policies the FDP is not static and may need to change on a day-by-day basis depending upon new vulnerabilities which arise. For example, Java was considered to be a "great invention" and the industry was assured by Sun Microsystems that it was not a security risk. Therefore as browsers evolved to become Java aware, Java applets were simply allowed through firewalls without restriction. It is likely that Java never appeared in any company's firewall policy as it was probably considered to be part of the WWW. Other examples of policy maintenance include changes to filtering rules of a network and rule changes resulting from the introduction of new services.

Review of the FDP

It is most important that the FDP remain under constant review to ensure that the policy reflects the current situation. As a result of FDP maintenance, the original policy can become unrepresentative of reality which can introduce security holes. Examples include — change of the systems expert, wrong versions of software being loaded following a system crash, etc. In many of these cases problems may not be detected until *after* a security breach has occurred.

4.6 Summary

This Chapter has provided an overview of the concepts and technologies that contribute to firewall architectures. It has also used the OSI model to provide a clearer abstraction of the functionality offered by firewall architectures.

In addition to functionality, the boundaries that define a firewall architecture have also been defined. The purpose of a firewall architecture is to reduce the zone-of-risk that an organisation is subjected to. While the security perimeter is important because it logically separates an organisations network(s) into trusted and untrusted domains. This Chapter has viewed internal networks as being contained within the domain of an organisation where they are inherently "trusted". In contrast external networks have been viewed as "untrusted" networks outside the control of any organisation. However, this view of internal and external networks is equally applicable within an organisation. For instance, a firewall may be used to separate sub-networks which are characterised by differing security policies, and possibly incompatible trust relationships.

Defining boundaries is important because it helps to clarify the extent of the security provided by a firewall architecture, and it defines logically and physically the relationship between internal and external networks. This is useful from the perspective of network security policy design which must be comprehensive and address the perceived risks, while satisfying the users expectations by allowing them to perform their daily tasks unimpeded.

The NSAP is particularly important as it defines explicitly the services which are permitted and those which are prohibited. Obviously, the most secure and manageable philosophy for NSAP design is to prohibit everything not expressly permitted. Implementation details of the NSAP are reserved for the FDP which defines *how* the firewall will restrict access to services. Both documents are a necessary part of the overall security policy, without the NSAP and FAP (or similar) it is hard to imagine how a firewall architecture could be selected and implemented while successfully considering all of the security implications.

At the end-of-the-day the most important advice for firewall security policy implementation and acceptance is that it must involve the end-users, and the NSAP at least must be distributed to all users to have the desired effect — comprehensive network security.

Chapter 5. Firewall Architectures

5.1 Introduction

A firewall architecture consists of a number of components which can be combined to provide increasing levels of network protection. Figure 5-1, compares the cost of various firewall architectures with the level of security they can be expected to provide.

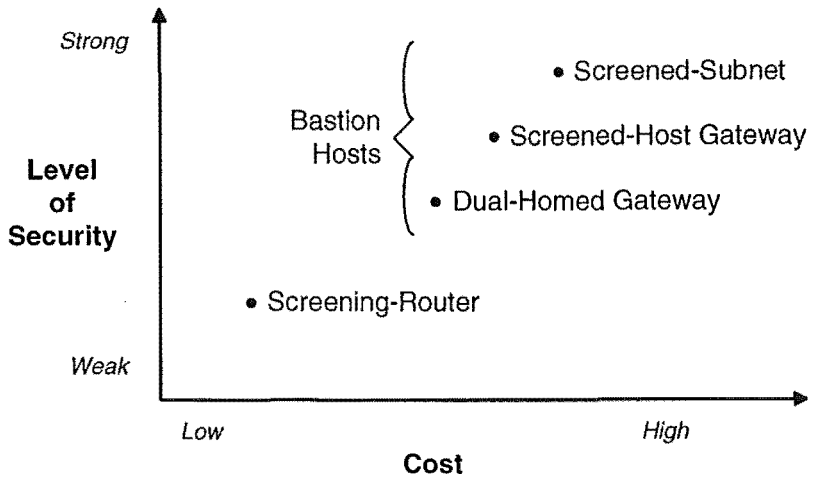

Figure 5-1 The cost of firewall architectures in comparison to the level of security they provide.

The following Sections describe the basic firewall architectures, as it is possible combine firewall components in a number of ways.

5.2 Screening-Router

A screening-router is a basic component of most firewall architectures, and usually consists of a commercial router. In some cases routing can be host-based, particularly on hosts using the UNIX operating system. Screening-routers filter the datagrams passing between the network connections in accordance with a previously defined routing table. Filtering is usually done on IP datagrams based on some, or all of the following fields:

- source IP address
- destination IP address
- TCP/UDP source port
- TCP/UDP destination port.

Additionally, some routers are able to distinguish which network interface a datagram arrives on, and use this information to decide how it should be filtered. This is particularly useful when traffic needs to be segmented from specific networks, and in eliminating IP source address spoofing. Datagrams which arrive at the external interface are known as *inbound packets*, while datagrams arriving at the internal interface are known as *outbound packets*.

The layers of the OSI model at which the screening-router normally functions are shown in Figure 5-2. The greyed in boxes of the protocol stack indicate the layers on which the filtering rules generally operate. The double headed arrow indicates the flow of traffic as it passes between the internal and external interfaces.

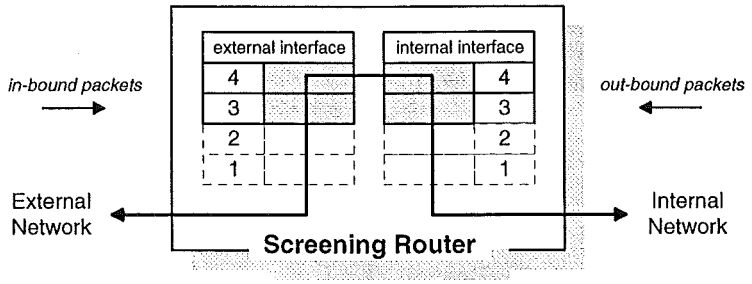

Figure 5-2 The OSI layers at which the screening-router functions.

Due to the inexpensive nature of screening-routers, they have been used in many networks as the sole component of the firewall architecture. Usually there are direct communication paths between multiple hosts on the internal and external networks (e.g. the Internet). In normal operation the zone-of-risk the internal network is exposed to is directly proportional to the number of hosts on the internal network, and the number of peer-to-peer connections to the external network. As the number of hosts and connections grow it becomes impossible to identify all possible threats should the router be compromised.

Figure 5-3 shows a simple router based firewall architecture, and differentiates between the external and internal networks.

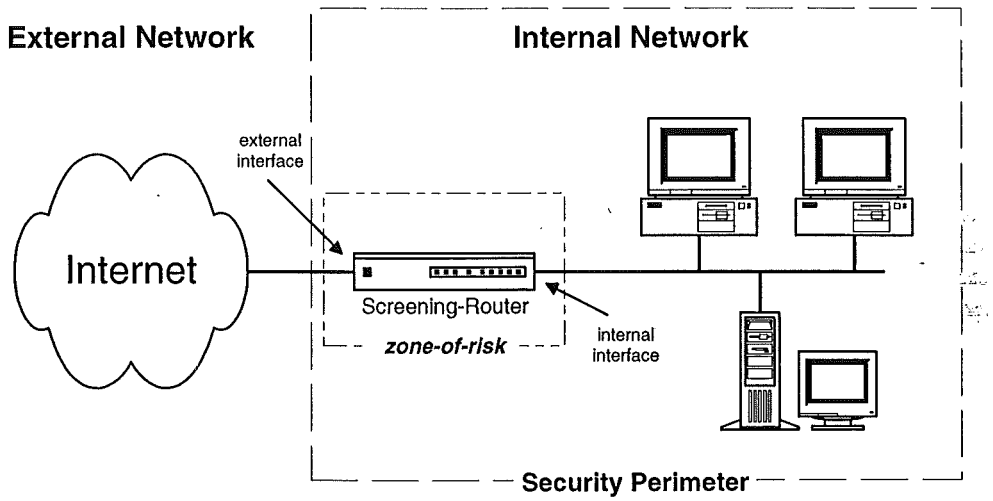

Figure 5-3 Typical screening-router based firewall architecture.

Routers are generally used to block connections from or to specific hosts or networks, and to block connections to specific ports. The ability to filter on both TCP and UDP ports adds considerable flexibility in defining security policies. As it allows the router to control which TCP services can be accessed, e.g. Telnet, FTP, SMTP, Finger etc.

Filter rules are generally specified using a table of conditions and actions which are applied to each datagram until a decision to route or drop is reached. If a datagram meets all of the conditions specified in the row of the table, the action specified in that row is carried out. Some systems apply the rules in a systematic manner from first to last. While others enforce an order based on the criteria in the rules, such as source and destination address.

As a simple example of the a screening-router, suppose that an organisation, *sprocket.com*, represented by the internal network shown in Figure 5-3, requires a mail connection to the organisation *widget.com*. A mail connection is characterised by a destination port number of 25, and a source port ≥ 1024. The sprocket.com mail connection is identified by the 2-tuple <IP number = 202.20.20.10, port ≥ 1024>, while the widget.com mail gateway is identified by the 2-tuple <IP number = 192.10.10.5, port = 25>. The (simplified) routing table required to permit this type of connection is shown below in Table 5-1.

Table 5-1 Example of a simple routing table.

Action	Source IP address	Port Number	Destination IP address	Port Number	Comment
allow	202.20.20.10	≥ 1024	192.10.10.5	25	Connection from sprocket.com to the widget.com mail gateway.
allow	192.10.10.5	25	202.20.20.10	≥ 1024	Allow replies from the mail gateway at widget.com to sprocket.com.
deny	*	*	*	*	If neither of the above rules are met deny access to all other datagrams.

The following are problems associated with many router implementations:

- *Configuration Control* – Perhaps the biggest draw back for sole use of a packet-level firewall in protecting a network is the amount of effort required for configuration control. Routers are difficult to program, every permitted combination of IP connection needs to be entered and in the correct order.

- *IP Spoofing* – Most filtering implementations rely on the accuracy of IP source addresses to make filtering decisions. However, IP spoofing takes advantage of the ease at which the source addresses can be faked. This is a case where the ability to filter on inbound datagrams is useful. To do this the router must make layer 1 and 2 (OSI model) information available for use in the filtering rules. If a datagram arrives on the external interface and its source and destination IP addresses are from the internal network then an IP spoofing attack is underway. Security of the internal network can be greatly improved if the router can be told to deny such datagrams.

- *Lack of Flexibility* – A major problem with packet filtering firewalls is their inability to administer fine grained access control at the user level. In particular a router cannot allow user A to access a service, while denying user B. If the service were blocked then datagrams generated by both A and B would be blocked.

5.3 Dual-Homed Gateway

The dual-homed (or multi-homed) gateway is a common and easily implemented Application-level firewall architecture. A dual-homed gateway is a host machine which has two network connection ports; one connected to the external network and one connected to the internal network (a multi-homed gateway simply has two or more network interfaces). With IP forwarding disabled a complete block of traffic between the two networks is ensured.

There are two ways a user can access the external network via the internal network. The first is by direct logon to the dual-homed gateway. This is not advisable as it makes the dual-homed gateway directly vulnerable to password cracking, and provides access directly to the firewall through software vulnerabilities, such as bugs or compilers being present on the host.

The second, and safest way for a connection to be made is through the Application-layer using a proxy server. A proxy server is an application which routes IP traffic from one port to another. Such an application can provide additional security mechanisms, such as user authentication, auditing, and logging facilities. These features are a great improvement over screening-routers which generally provide no more than rudimentary facilities.

The problem with using proxy servers is that they usually have to be written for each service that is offered. However, basic proxy servers for standard TCP/IP services, such as Telnet, FTP, WWW, etc., are generally available for UNIX and Microsoft Windows (i.e. Windows 95 and Windows NT) environments.

A dual-homed gateway will generally perform the same packet-level functions as the screening-router. It may also provide additional functions such as address translation, and IP masquerading. This is shown in Figure 5-4, which relates the dual-homed firewall architecture to the OSI model.

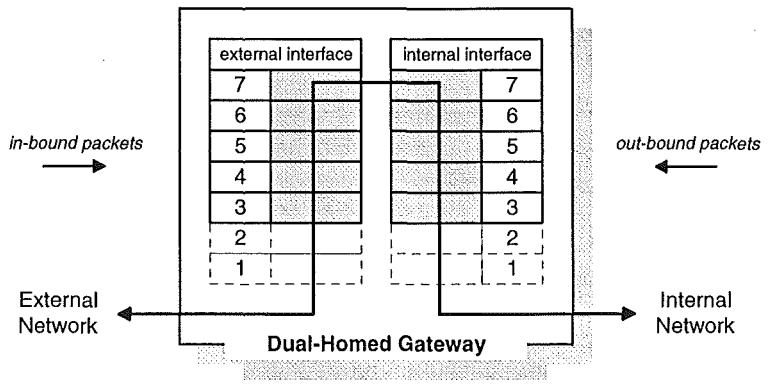

Figure 5-4 The OSI layers at which the dual-homed gateway functions.

Figure 5-5 shows a simple dual-homed firewall architecture, and differentiates between the external and internal networks.

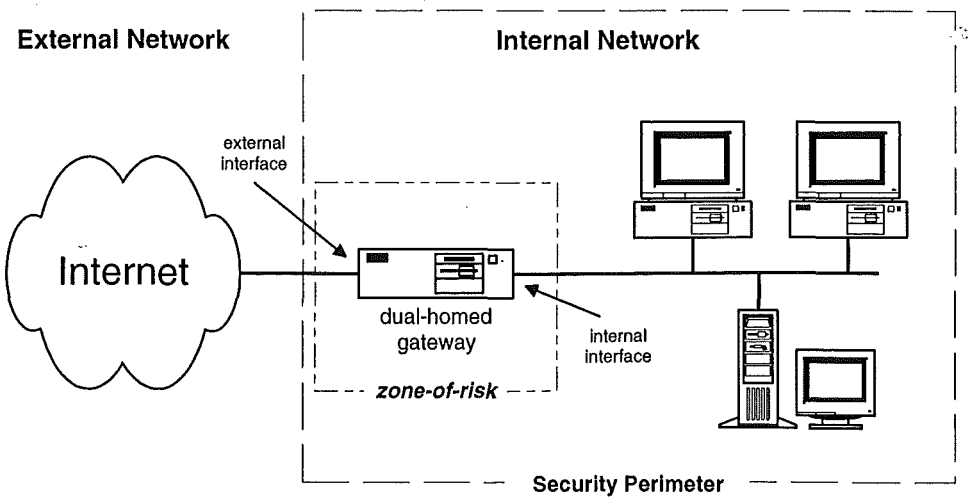

Figure 5-5 Typical dual-homed gateway.

The greatest threat to the security of a dual-homed gateway is when an attacker gains login access to it. As stated above, allowing users to log directly onto the bastion-host makes it easier for an attacker to gain a foothold on the machine. Therefore login should always be through application proxies on the dual home gateway.

If an attacker obtains login access to the dual-homed gateway the internal network is subject to intrusions. The zone-of-risk has been extended from the dual-homed host to include the entire internal network. The following is a list of sources from which an attack can be mounted [Hare et al., 1996]:

- Weak permissions on the file system.
- Internal network NFS-mounted volumes.
- Permissions granted to Berkley r-utilities (e.g. *rlogin*) through host equivalent files, such as the rhosts file (see Section 3.2.2), often found in user home directories which have been compromised.
- Network backup programs that could restore excessive permissions.
- The use of administrative shell scripts that have not been properly secured.
- Learning about the system from older software revision levels and release notes that have not been properly secured.
- Installing older operating system kernels that have IP forwarding enabled.

The key for the attacker is to gain enough system privileges to be able to change the UNIX kernel variable *ipforwarding*, which controls IP forwarding. Once this variable has been enabled the firewall has been completely subverted.

There are a number of aspects, apart from disabling IP forwarding, which can be checked to ensure the security of a dual-homed gateway; the following list is adapted from [Hare et al., 1996]:

- Remove all programming tools; including compilers, linkers, utilities, and services not specifically required for the operation of the dual-homed gateway.
- Ensure programs that have SUID and SGID permissions, and if not required are removed. Check that no excessive permissions on files and programs exist.
- Use disk partitions so that denial-of-service attacks designed to fill all available disk space on a partition are confined to that partition.
- Remove unneeded system and special accounts, e.g. disable guest accounts, and maintenance accounts found on some proprietary systems (including screening-routers).
- Delete network services that are not required.

5.4 Screened-Host Gateway

The screened-host gateway is implemented using a screening-router and a bastion-host. It is one of the most popular firewall architectures. The bastion-host is usually placed on the internal network, with the screening-router configured such that the bastion-host is the only machine reachable from the Internet. To restrict Internet access further the screening-router is generally configured to block all traffic not destined to specifically authorised ports on the bastion-host. This has the effect of controlling the number of available services.

The combination of screening-router and bastion-host means that the screened-host firewall architecture effectively functions from layer 3 (or layer 2 if the screening-router can filter datagrams based on the network interface they arrive on) to layer 7 of the OSI model.

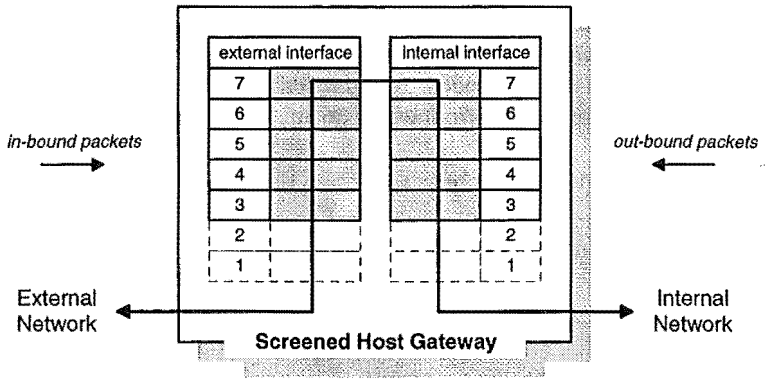

Figure 5-6 The OSI layers at which the screened-host firewall architecture functions.

Major benefits of screened-host gateways include reduction of router programming complexity, and improved connectivity for local users. As all traffic is passed through one single point i.e. the bastion-host, then the rules for configuring the router table need only consider the bastion-hosts IP address. All other datagrams arriving at the inbound or outbound ports of the screening-router can be discarded, which greatly simplifies the required packet filter rules. Figure 5-7 shows a typical screened-host firewall architecture.

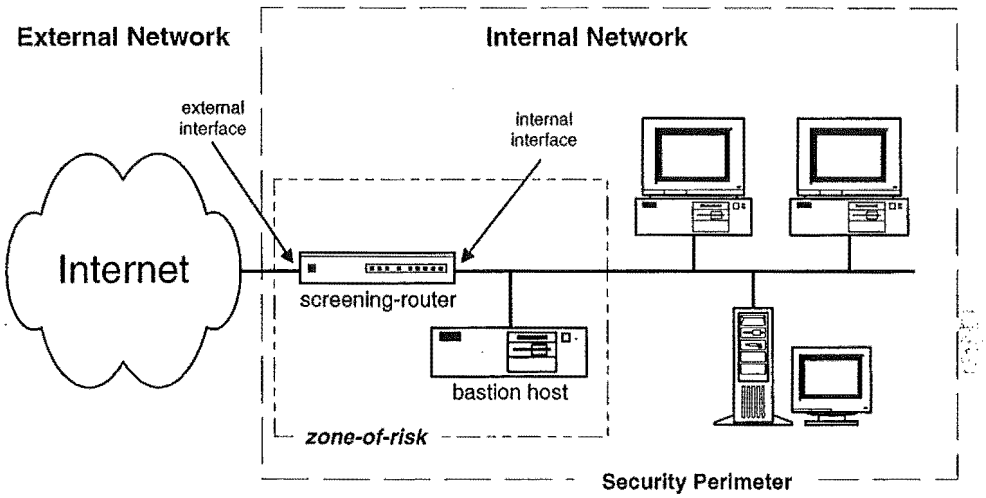

Figure 5-7 Typical screened-host firewall architecture.

If the internal network is a virtual local area network configuration with no subnets or additional routing. Then the screened-host gateway can be implemented without changes to the original LAN. Users then have the ability to connect directly through the bastion-host to the external network, without excessive routing overhead.

The zone-of-risk incorporates only the screening-router and bastion-host. The security of this firewall architecture is determined by the accuracy of the packet filter rules in relation to the security policy, and

the level of assurance regarding the software running on the bastion-host. If an attacker gains entry to the bastion-host then the threats to the internal network are similar to those of the dual-homed gateway.

There is a major problem with the architecture of the screened-host gateway described above. The problem is inherent with positioning the bastion-host on the internal network, and relying on the screening-router to control the traffic flow to and from it. A potential Achilles heal exists with the compromise of the screening-router. If this happens, either through, miss-configuration of the packet filtering rules, or through an attacker gaining access to the screening-router via a proprietary maintenance account, then the entire internal network is at risk. In effect compromising the screening-router effectively subverts the bastion-host. Once an attacker has control of the screening-router, all traffic can be routed to the external network.

A more secure implementation is to use a screening-router connected to a dual-homed gateway. This architecture ensures that the bastion-host is not circumvented if the screening-router is compromised. The attacker has to overcome the dual-homed gateway before the internal network is at risk. Of course, this architecture offers no improvement if an attacker is able to enter through the screening-router and compromises the bastion-host directly.

5.5 Screened-Subnet

A screened-subnet firewall architecture, consists of an isolated network known as the *exterior network*, positioned between the external and internal networks. This configuration allows non-critical hosts, such as WWW-servers and anonymous FTP sites, to be placed on the exterior network. The advantage of removing these servers from the internal network is realised when one is compromised. As they have no connection with the internal network a compromise does not directly effect the safety of the internal network — although the compromised host could be used to launch attacks, for example an attacker could use it to view all traffic between the internal and external networks. The servers also benefit from the protection afforded by the external screening-router. Bastion-hosts are placed on the exterior network to provide interactive terminal sessions, or application-level firewalls [Ranum, 1996]. The combination of firewall components means that the screened-host firewall architecture functions form layer 3 (or layer 2 if the screening-router(s) can filter datagrams based on the network interface they arrive on) to layer 7 of the OSI model (see Figure 5-8). The screened-subnet is generally considered to be the most secure firewall architecture.

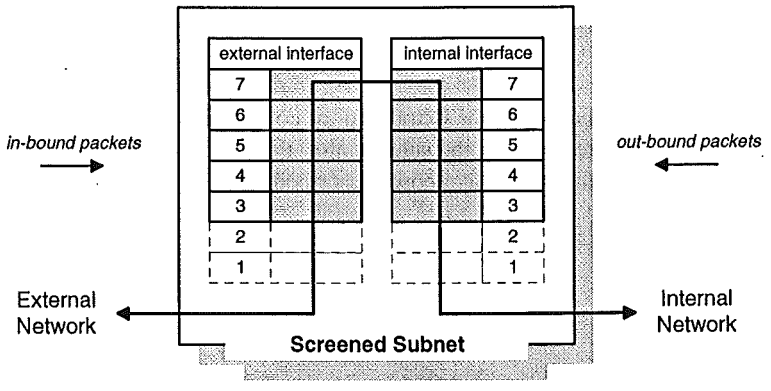

Figure 5-8 The OSI layers at which the screened-subnet firewall architecture functions.

The bastion-host provides the sole point of access to machines on the internal network, and forces all services through the firewall to be provided by application proxies or circuits. Protecting the bastion-host are two screening-routers, one between the external network and subnet (known as the external router), the other between the subnet and internal network (known as the internal router). Therefore the

zone-of-risk for this configuration consists of only the two routers, and the bastion-host, as well as any other hosts placed on the subnet. An example of this configuration is shown in Figure 5-9.

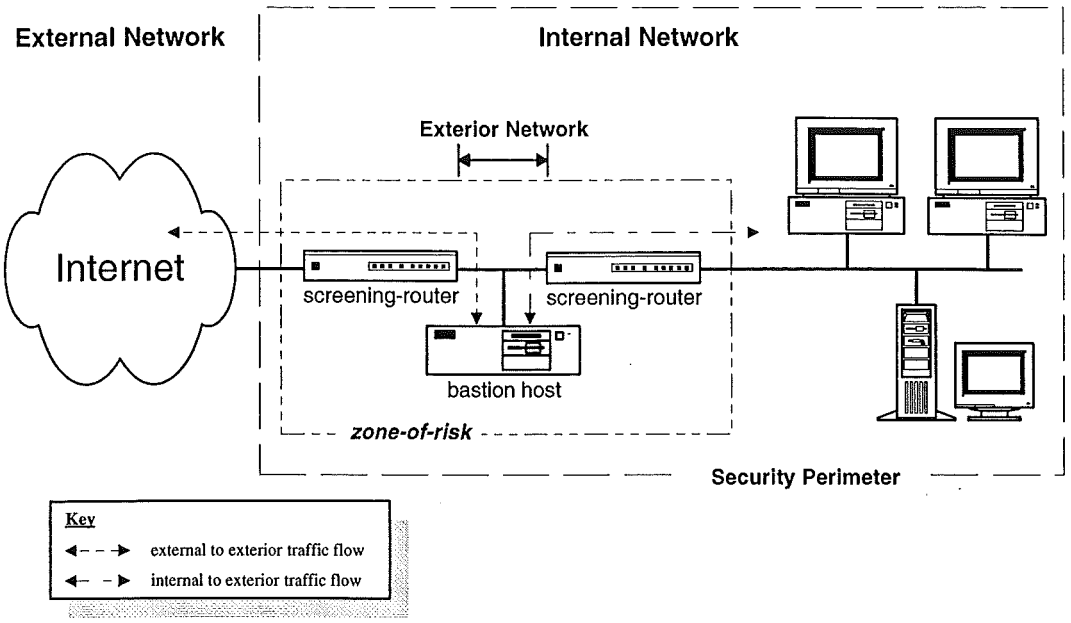

Figure 5-9 Typical screened-subnet firewall architecture.

The strength of this firewall architecture is derived from the fact that an attacker must, in general, subvert the external router followed by the bastion-host and finally the internal router. If the screening-routers are configured so they cannot be managed remotely from the network, then subverting the bastion-host without setting off alarms and appearing in audit logs would be very difficult.

As with the screened-host gateway, if the screening-routers can be directly compromised by logging into them and reconfiguring their routing tables the bastion-host can be negated and the internal network put at risk. In fact, this is the reason why highly security conscious organisations often install screening-routers from different manufactures, to reduce the risk that a vulnerability in one router will be present in the other.

One draw back with this configuration is the extra level of complexity added to the definition of packet filter rules, especially if there are hosts on the subnet other than the bastion-host. The overall threats to this firewall architecture are the same as those described for the dual-homed and screened-host gateways.

5.6 Hybrid Gateways

The term "hybrid gateway" is normally used to describe any non-standard firewall architecture, it may be that the components are proprietary or that TCP/IP is not the predominant networking protocol. It is important that any belief that a non-standard hybrid gateway will provide "security through obscurity" should be dispelled. A hybrid firewall architecture may slow a determined attacker, but over a period of time they will build up enough information to understand the gateway and its security mechanisms.

5.7 Firewall Limitations

A firewall architecture is a powerful tool for network security, but it is not a panacea for all ills. Firewalls are particularly suited to controlling access to services and network resources, and monitoring and auditing traffic travelling between the networks that it connects. It is important to understand that a firewall architecture's influence is localised. The analogy of a firewall being like a castle wall holds here — the wall may protect those behind it, but affords no protection to those people or hamlets on the other side! Therefore, it may not be enough to simply build a single castle, especially when building Extranets and Intranets which move information and resources outside the protective influence of a firewall. However, this problem is the topic of Chapter 7 which looks at cryptography and VPNs as a means of extending the protection offered by firewall architectures or gateways to incorporate untrusted networks, such as the Internet.

Firewall architectures are focused on protecting against attacks launched at the lower layers of the TCP/IP suite, in particular the Network and Transport-layers. For example, hosts behind a screening-router configured to drop source-routed IP datagrams are virtually immune to IP spoofing attacks (see Chapter 3 for a discussion of common threats to the TCP/IP suite). The screening-router simply discards any IP datagrams with this feature enabled.

In contrast, firewall architectures provide almost no protection against problems with higher layer protocols, unless they are performing some type of analysis (or filtering) of the Application-layer data contained within the TCP segments or UDP datagrams (or other Transport-layer protocols). The most sophisticated TCP-based proxy provides no protection at all if a server it protects contains exploitable vulnerabilities.

A recent problem with Microsoft Windows based WWW-servers[24] highlights the difficulties that firewall architectures have in dealing with security at the Application-layer. All 32-bit Microsoft Windows operating systems associate two different filenames with a stored file, a short name of 8 characters and a long name which must be less than 255 characters — the short filename is derived from the long name in a predictable manner. For example, the long filename "Abcdefghijk.xyz" is also represented with the short filename "Abcdef~1.xyz".

Vulnerable WWW-servers attempted to restrict access by building an internal list of restricted filenames. However, for files with long names, only the long and not the short filename was added to the internal list. This leaves the file unprotected by the WWW-server because the file is still accessible via the short filename. Attackers could take advantage of this exploitable vulnerability to gain unauthorised access to files protected solely by the WWW-server. Unless the firewall had been configured with a filter to deal with this problem requests for protected files would succeed — the firewall architecture provides no protection whatsoever.

A related and equally difficult problem to deal with is malicious content that is accessed legitimately by users behind the firewall. Malicious content includes virus infected software and data files (e.g. infected with macro-viruses), Trojan horse programs, and executable content (e.g. Java applets, ActiveX controls, etc.). There are several common ways such malicious content can pass through a firewall:

- Email attachments
- File transfer (e.g. FTP)
- WWW-browsing

The problem of malicious content leads to the question *"what protection should a firewall architecture provide?"* The uninitiated usually expect firewalls to provide protection against all threats including malicious content. Unfortunately, this is nearly always not the case and in fact it is difficult to see how a firewall could provide such omnipresent protection. However, some firewall architectures do attempt

[24] See CERT Advisories CA-98.04, February 11, 1998; which is available at **http://www.cert.org/advisories/CA-98.04.Win32.WebServers.html**

such a feat by providing virus scanners that automatically check attachments as they are received, and some particularly "advanced" firewalls provide scanners that check for malicious Java applets.

Placing too much confidence in such "alchemy" could prove disastrous. For example, how can a virus scanner check email attachments if the attachment is instead encoded (e.g. using UUencode) within the message body? (This is still a fairly common practice.) Even if this was not a major problem, consider the performance impact (in fact virus scanners by default only scan for the most common viruses) that scanning every email attachment using every possible encoding format would have — some of the most common encoding formats include; TAR, ZIP, BinHex, ARJ, ARC, LZH, Base64, UUencode, XXencode — and what about recursively encoded files? Scanning for malicious Java applets is perhaps even more pointless because there is an infinite number of ways a program can be written or slightly altered to prevent the scanner from recognising it.

Although virus and malicious content scanners are imperfect they do play an important role in preventative computer security, and should not be discounted entirely as useful additions to the Application-layer security of firewall architectures — as long as it is recognised that they are fallible. It should also be remembered that scanners are a "reactive" technology, by their very nature they will always lag behind the development of new viruses and malicious content. Therefore, new variations will nearly always avoid detection.

Obviously, new methods must be found to deal with the problem of malicious content. Perhaps the most promising method involves the use of digital signatures and public-key certificates (see Sections 7.3.3 and 7.3.4 respectively). The use of these mechanisms provides a way of associating a measure of "trust" to the supplier of software or data files.

The problems with deficient Application-layer services, viruses, and malicious content, highlight the fact that in some cases a well defined security policy can offer more defence than can technical solutions. For example the NSAP and FAP could restrict the use of Java applets and ActiveX controls to those that have recognised digital signatures that are trusted by the organisation. The policy could then be enforced by a firewall architecture that can be configuring to check and permit only trusted executable content, while scanning could also be used to provide an additional safety check.

Unfortunately, firewall architectures are more often designed to "keep people out" rather than to "keep people in". Most firewall architectures provide internal users with uncontrolled access to external networks through services such as email, HTTP, FTP, Telnet, etc. — which is ironic considering the trouble that organisations go to to prevent abuse of these services by external attackers. Thus the majority of firewall configurations provide little protection against insider abuse, although some support advanced identification and authentication techniques that are usually based on the cryptographic mechanisms and protocols discussed in Section 7.3.

The only alternative is to incorporate intrusion detection mechanisms within computer systems and networks that are capable of detecting insider abuse. Unfortunately, such systems have not become widespread and are themselves plagued by many problems. So until effective safeguards against insider abuse are found it will remain relatively easy to abuse the trust given to users behind the firewall.

5.8 Summary

The firewall architectures discussed in this Chapter represent the fundamental configurations used to control the flow of traffic between networks that have different security policies or levels of trustworthiness.

In particular, firewalls cannot protect against installation errors or software bugs, nor can they protect against malicious programs imported through legitimate means. A firewall can only defend against known security threats, and will always be vulnerable to new ones. Of course the only truly secure configuration is to block all applications such as email, FTP, and executable content — although this may upset employees that have come to rely on such services.

Chapter 6. Certification of Firewall Technology

6.1 Introduction

There are many firewall products available which implement the various architectures described in Chapter 5, all of which can differ in price, performance, effectiveness, and quality. The firewall architecture that an organisation implements should be determined in consultation with its security policy documents. However, the most important aspect to consider is the level of assurance that the given architecture provides. To date, assurance has been attributed to firewall products through independent evaluation using some suitable criteria.

Currently, there are three criteria being used to evaluate firewall products. The first two have been developed for government certification programmes, and are known as the *Information Technology Security Evaluation Criteria* (ITSEC), and the *Common Criteria for Information Technology Security Evaluation* or simply the *Common Criteria* (CC). The final criteria was developed for a commercially driven programme run by the *International Computer Security Association* (ICSA), and is known as the *FWPD*[25] *Criteria* (FWPDC).

Certification of firewall products has taken two distinct paths:

- *Government Certification* – uses recognised, formal evaluation criteria such as the ITSEC and the CC, and is aimed at meeting the needs of public organisations such as the government, military, and law enforcement. Such organisations often require IT security products and systems suitable for processing and protecting information that ranges from unclassified to nationally classified (e.g. CONFIDENTIAL, SECRET, TOP-SECRET). Obviously, systems which handle classified information must prove their trustworthiness to an acceptable level. Both the ITSEC and CC require that all aspects of a product or system be reviewed and investigated to an extent commensurate with the claimed level of trustworthiness. The impetus of government certification schemes is to provide governments with a trustworthy range of *commercial-off-the-shelf* (COTS) IT security products.

- *Commercial Certification* – is aimed at meeting the needs of private organisations, in a timely and cost-effective manner. Private organisations rarely (if ever) handle nationally classified information (this may also be true of many public organisations). Instead, they may be required to process information which requires protection to a level that satisfies their legal obligations, e.g. New Zealand's Privacy Act (1993). In addition they may also wish to protect information that is commercially sensitive, the compromise of which may have a detrimental impact on the organisation's competitiveness or ultimate survival. In general, private organisations require certification that shows they meet accepted industry standards. The FWPDC provides this by focusing evaluation towards firewall testing, in particular *penetration testing*. Therefore ICSA certification is an assurance that the firewall is able to withstand attacks based on *current* threats and vulnerabilities. The ICSA is currently the only commercial organisation offering this type of certification.

Regardless of the criteria used the outcome of a successful evaluation is the award of a *certificate*. From the perspective of developers and vendors a certificate can lead to competitive advantage and market penetration. While from the customers perspective a certificate allows a defined level of trust to be placed in the IT security product or system.

[25] FWPD is an acronym for the Firewall Product Developers' Consortium which was established in 1995 to provide a forum in which developers of competing firewall products could work toward common goals, such as certification.

6.2 Problems with Firewall Evaluation

Firewall technology has advanced considerably in the past four years, yet the practice of firewall evaluation has not kept pace [Schultz, 1996]. The problem is that firewalls are unlike traditional security products which are designed to counter the known threats of a particular operational environment, and have well defined modes of operation. Perhaps the most important distinction is that traditional security products tend to evolve slowly. The culmination of these factors mean there is less requirement for modification. For evaluated products this means re-evaluation or review, which is necessary to determine whether the modification has impacted on the products security objectives.

Firewall architectures on the other-hand consist of many components, such as an operating system, management tools, and proxy servers. They can also have a very wide range of operating modes and configurations, for example a firewall may support encrypted connections and allow user defined services. All of these aspects must be considered and evaluated as a whole. However, the greatest problem with firewall certification is the speed at which the technology evolves.

Each time a firewall is modified to support a new service, or provide additional functionality, such as Java, ActiveX, or SSL, it must be re-evaluated, or at the least reviewed. This is necessary to assess the impact that the modification may have had on the various components contributing to the firewall architecture. It is very possible for a modification in one component to introduce a vulnerability in another. Unfortunately, re-evaluation can be as expensive as the initial evaluation. Evaluation, re-evaluation, and review all introduce delays in the marketing of a firewall product. This impacts directly on the sponsor because they are unable to recover their investment through selling the certified firewall until it has officially received its certificate.

6.3 Government Certification

A number of national governments have invested in programmes to evaluate Information Technology security products and systems. The majority of these schemes have been developed with the objective of meeting the needs of government and industry for IT security evaluation and providing a basis for international mutual recognition of evaluation certificates. As previously stated, the ITSEC and CC are currently the only criteria being used to evaluate firewall products. It is important to note that the ITSEC and CC were designed to enable the evaluation of any IT security product or system.

It is not possible to review all of the national schemes and programmes which currently subscribe to the ITSEC or CC. Instead the following Sections focus on the ITSEC and its application by the *Australian Information Security Evaluation Programme* (AISEP). The CC is not discussed in detail as it is still a draft document, and has only been used in trial evaluations [CCITSE, 1996].

6.3.1 Development of Information Technology Security Evaluation Criteria

In 1978 the United States Department of Defence in conjunction with the MITRE Corporation began working on a set of computer security evaluation criteria that could be used to assess the degree of trust one could place in a computer system used to protect classified data. This work led to the development to the *Trusted Computer System Evaluation Criteria* (TCSEC), or otherwise known as "The Orange Book," due to the colour of its cover. The TCSEC is limited to evaluating the effectiveness of security controls built into automatic data processing system products [TCSEC, 1985]. In effect this means the operating system and the hardware on which it operates. As the functional requirements specified in the TCSEC are applicable only to standalone operating systems, it cannot be used by itself for example to evaluate a firewall which operates in a networked environment. Due to the specific nature of the TCSEC, variations (known as interpretations) have been developed to evaluate different types of IT security products and systems. For example a *Trusted Network Interpretation* (TNI) of the TCSEC, also referred to as "The Red Book," has been developed. This is a restating of the requirements of the TCSEC in a network context [TPEP, 1998]. The collection of these interpretation documents along with

a number of guidance[26] documents (e.g. Guide to understanding Mandatory Access Control, and Password Guidelines) is known as the Rainbow Series — each document has a distinctly coloured cover. In an effort to replace the TCSEC with a criteria more in-line with the ITSEC, the *Federal Criteria* (FC) was developed. A draft version was released for public comment in December 1992, however, this effort was overtaken by the CC and the FC never progressed beyond the draft stage.

A number of other countries, mostly European, also have significant experience in IT security evaluation and have developed their own criteria. France, Germany, the Netherlands and the United Kingdom recognised that a significant amount of work remained to be done in the area of IT security evaluation. Therefore, it was decided that the efforts of these four countries be combined to produce a common, harmonised criteria. The best features from each national criteria were combined to form the harmonised *Information Technology Security Evaluation Criteria* (ITSEC) [ITSEC, 1991].

As well as the Americans and Europeans, the Canadians developed the *Canadian Trusted Computer Product Evaluation Criteria* (CTCPEC) which is the equivalent of the TCSEC. It is more flexible than the TCSEC, having aspects of the ITSEC, but still maintains a close compatibility with individual TCSEC requirements.

In June 1993, the authors of the CTCPEC, FC, TCSEC, and ITSEC combined their efforts and began a project to align their criteria and create a single draft of the CC. The intent of this ongoing project is to resolve the conceptual and technical differences found in the source criteria, and then deliver the results to the ISO as a contribution toward its work in progressing an international standard for a general IT security evaluation criteria [CCITSE, 1996]. An initial draft (version 1.0) was released in January of 1996, but is expected to be replaced in early 1998 with version 2.0[27].

The primary purpose of these criteria is to provide government and industry with a metric to measure the level of "trustworthiness" which can be placed in a product or system. It also provides developers with guidance in the development of secure IT products and systems, especially those intended for the processing of sensitive or classified information. As previously mentioned, a major desire and perhaps realisable with the CC is international mutual recognition. Mutual recognition would, for example, allow the evaluation certificate for the *Cyberguard Firewall* evaluated under the UK ITSEC scheme [CPL, 1997] to be recognised by the AISEP, which also uses the ITSEC. Unfortunately, to date, political differences and difficulties in equating the outcomes from the different certification schemes have prevented this.

6.3.2 Overview of the ITSEC

The ITSEC is a standardised criteria with a formalised methodology, known as the *Information Technology Security Evaluation Methodology* (ITSEM). The ITSEM provides guidance on how to interpret and apply the ITSEC. Its purpose is similar to the Interpreted TCSEC[28], and the *Common Evaluation Methodology* (CEM) [CEM, 1997] developed for the CC.

The ITSEC places emphasis on integrity and availability, and attempts to provide a uniform approach to the evaluation of both products and systems. The ITSEC makes a distinction between how well the product provides security in the context of its actual or proposed operational use — *effectiveness*); and whether it achieves the stated security objectives and features — *correctness*. By taking this approach, the ITSEC allows less restricted collections of requirements for a product. However, this is arguably at the expense of more complex and less comparable ratings, as well as the need to carry out effectiveness analysis of the features claimed for the evaluation.

---

[26] The guidance documents, which are part of the Rainbow Series, expand and clarify the requirements of the TCSEC and the various interpretations (e.g. TNI). They offer guidance only, the TCSEC or the interpretations provide the only metric for evaluation.

[27] An unofficial release of the CCITSE version 2.0 is available at **http://csrc.nist.gov/cc/ccv1x/wip_list.htm**

[28] The Interpreted TCSEC is available at **http://www.radium.ncsc.mil/tpep/library/tpep/ITCSEC.ps**

The ITSEC defines seven assurance levels; E0, E1, E2, E3, E4, E5, and E6. E0 is reserved for products which fail evaluation, while E1 to E6 represent increasing assurance (see Figure 6-1). Descriptions of each level are given in Appendix A.

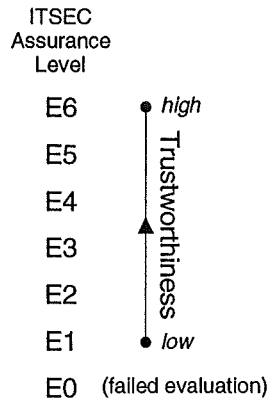

Figure 6-1 ITSEC Assurance Levels.

The time taken to evaluate a product increases with each of the assurance levels because more work needs to be completed by the evaluator. The following time-scales from [UKSP04, 1996] provide a general indication of the time taken to have a product evaluated under the UK ITSEC scheme. These time-scales represent the elapsed time from the start of a product evaluation to delivery of the *Evaluation Technical Report*[29] (ETR) to the *Certification Body*[30]:

- E1 – up to 6 months
- E2 – 6 to 18 months
- E3 – 8 to 24 months

Estimates for the time-scales of assurance levels greater than E3, i.e. E4, E5, and E6, are not indicated by [UKSP04, 1996]. This is perhaps due to the smaller number of evaluations completed at these levels. However, due to their complexity and the greater amount of work required, assurance levels of E4 and higher could be expected to take several years to complete. The AISEP does not publish time-scales however it would be expected that they are similar to those given above.

The CC has a structure which is very close to the ITSEC, but includes the new concept of a *Protection Profile* (PP). The PP permits the implementation independent definition of security requirements for a set of products or systems which complies fully with a set of security objectives. A PP can be developed by user communities, IT product developers, or any other interested parties, which defines a set of common security requirements. In fact the initial draft release of the CC included a predefined PP for packet-level firewalls. Subsequently, additional work has been done in developing more applicable and effective PPs[31].

[29] The ETR is a report produced by an evaluation facility which details the findings of an evaluation and forms the basis of the certification process for a product or system.

[30] The Certification Body is an independent and impartial national organisation in the UK that performs certification. It is managed by CESG and the *Department of Trade and Industry* (DTI), and is responsible for the day-to-day running of the UK ITSEC Scheme, certifying evaluation results, and licensing evaluation facilities. The Certification Body is under the direction of the Management Board which is responsible for setting policy and for overseeing its implementation [UKSP04, 1996].

[31] Additional work on PPs can be found at **http://csrc.nist.gov/cc/pp/pplist.htm**

The CC has seven assurance levels, EAL1 to EAL7, where EAL2 to EAL7 correspond very closely to ITSEC E1 to E6 (see Figure 6-2). It is interesting to note that the CC has no concept of a failed evaluation (i.e. EAL0) unlike the ITSEC which denotes this as E0. An evaluation is either successful and granted a CC rating, or it simply does not get a rating. This was done to prevent vendors using the zero rating as a marketing tool. It was thought that vendors would claim that they were granted a CC rating with the implication that the rating was a positive one when in reality the product failed evaluation [Cohen, 1998].

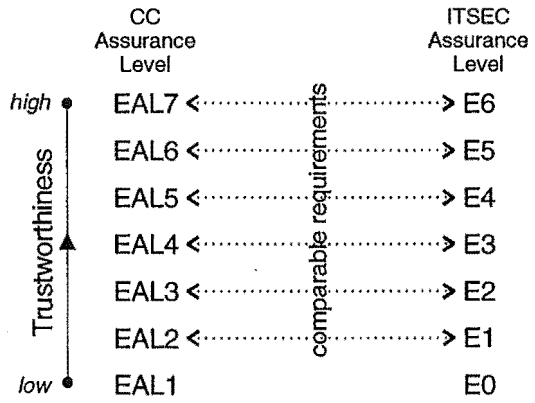

Figure 6-2 Comparison of CC and ITSEC assurance levels.

The new EAL1 assurance level is intended to allow the detection of obvious errors for minimal cost, and represents the minimum at which a meaningful assurance level can be awarded. EAL1 requires a minimum of documentary evidence, augmented with an appropriate level of ad hoc penetration testing. This approach is suitable for the detection of obvious security weaknesses only. The following quote from [CCITSE, 1996] provides an insight into the reasoning behind the inclusion of EAL1;

> *"EAL1 is applicable in circumstances where those responsible for user data may wish or be obliged to seek independent assurances in the IT security, but the risks to security are not viewed as serious. Under these circumstances, an EAL1 rating would be of value to support the contention that due care had been exercised with respect to personal or similar information."*

In effect an EAL1 evaluation would be very similar an FWPDC evaluation conducted by the ICSA (see Section 6.4) because both depend on penetration testing to provide a minimum acceptable level of trustworthiness. Commercial evaluation facilities which already provide evaluation services for government certification schemes could conduct EAL1 evaluations to compete with commercial certification schemes like that offered by the ICSA.

Perhaps in anticipation of competition for the evaluation of low assurance IT security products and systems, the UK ITSEC scheme has expanded their criteria to include the EAL1 assurance level for the evaluation of IT security systems. This is directed towards systems originally built without thought for evaluation and for which the documentary evidence required for higher assurance levels does not exist and would not be cost effective to produce [UKSP06, 1997].

6.3.3 Overview of the Australian Information Security Evaluation Programme

In June 1994 the *Defence Signals Directorate*[32] (DSD) announced the establishment of the *Australian*[33] *Information Security Evaluation Programme* (AISEP). As stated above the AISEP currently subscribes

[32] The Defence Signals Directorate has a similar role in Australia, as the Government Communications Security Bureau has in New Zealand. Both organisations are the national information security authority for their respective countries.

to the ITSEC, however the CC has been identified as the long term solution to problems such as mutual recognition. Transition from the ITSEC to the CC has already begun, although only trial evaluations using the CC will be undertaken in the near future.

The objective of the AISEP is to provide the Australian and New Zealand Governments and industry with an efficient and cost effective service in the evaluation and certification of IT security products and systems [EM1, 1997]. It offers evaluation and certification services to:

- developers – to enable them to demonstrate the security claims of their products; and
- users – to allow them to satisfy themselves and others that their systems actually meet their security objectives.

Independent third-party evaluation is carried out by *Australian Information Security Evaluation Facilities* (AISEFs). Currently two AISEFs are operating; Admiral Computing based in Canberra, and Computer Sciences Corporation (CSC) based in Canberra and Adelaide. Each have had to meet the requirements of the *National Association of Testing Authorities* (NATA) to be registered as a testing laboratory. In addition they must he found technically competent by the *Australian Certification Authority* (ACA) before the AISEP will issue a licence.

Evaluation begins with the Sponsor[34] developing a Security Target for their product or system. The Security Target is a specification of the security required of the Target of Evaluation[35] (TOE) and is used as a baseline for the evaluation. The security target specifies the security enforcing functions (SEFs) of the TOE. It also specifies the security objectives, the threats to those objectives, and any specific security mechanisms that will be employed. Most importantly it states the intended assurance level for the product.

Once developed the Security Target along with a number of other evaluation deliverables are passed to the AISEF. This commences the main part of the evaluation process where the evaluators perform the ITSEC evaluator actions, including penetration testing based on a list of potential vulnerabilities. All problems identified during this phase are discussed between the relevant parties, such as the sponsor, developer, or certification group. It is the responsibility of the sponsor to resolve all identified problems. If this is not possible, the sponsor may abandon the evaluation or accept potential limitations in the certificate/certification report. The major output of this phase is the ETR, which records the results of the evaluation work and identifies any unsolved problems. The ETR[36] is the basic input to the final phase of evaluation, known as the *certification process*.

During the certification process, the Certification Group[37] reviews the ETR to determine whether the security target is met by the TOE, taking account of any factors outside the scope of the evaluation such as the strength of cryptographic mechanisms. The Certification Group also confirms that the evaluation has been performed in accordance with the rules of the AISEP, and as agreed between the AISEF and the sponsor. The conclusions of the Certification Group, along with the assigned assurance level, are recorded in a certificate and certification report, which are passed to the sponsor and the AISEF. Copies of the certificate are distributed to various interested parties, including all AISEFs and specific government departments.

---

[33] In the near future the word "Australian" will be replaced by "Australasian", to reflect the involvement of New Zealand, and the AISEPs increasing recognition within Australasia.

[34] The "sponsor" is the person or organisation that requests an evaluation.

[35] The TOE is an IT system or product which is subjected to security evaluation.

[36] Since the ETR contains commercially and possibly nationally sensitive information, it is not a public document.

[37] The Certification Group (equivalent to the UK ITSEC Scheme's Certification Group) is an independent and impartial national organisation in Australia that performs certification. It is managed by DSD, and is responsible for the day-to-day running of the AISEP, certifying evaluation results, and licensing evaluation facilities. The Certification Group is under the direction of the Management Policy Board which is responsible for setting policy and for overseeing its implementation [EM7, 1997].

Table 6-1 lists the firewall products which have been certified or are currently in-evaluation with the AISEP. The UK ITSEC Scheme also has a number of certified and in-evaluation firewall products which are shown in Table 6-2 — it is interesting to note that some of the firewalls have CC assurance levels. The UK, US, and Canada have agreed to the *Interim Mutual Recognition Agreement* (IMR), which has been signed by the *National Security Agency* (NSA), the *Communications-Electronics Security Group* (CESG) in the UK, and the *Communications Security Establishment* (CSE) in Canada. The implication for vendors is that products evaluated with the CC need only be evaluated once as the resulting certificate is accepted by the other IMR participants. The Black Hole firewall version 3.01 E2 from Milkyway Networks Corporation has been evaluated to CC assurance level EAL3 by CSE, and is the first and currently only firewall product to be recognised by the IMR agreement. This evaluation took 12 months to complete due to a lack of an evaluation methodology, and the need to develop a security target as the existing predefined PPs were not sufficient. It is estimated that the evaluation would have taken 9 months if a suitable methodology had been available [Cohen, 1998].

From a New Zealand perspective, once a certificate has been received by the *Government Communications Security Bureau* (GCSB) the product is listed in the *Preferred Products List* (PPL). The PPL lists security products which have been successfully certified through recognised schemes and programmes, such as the AISEP. Government departments within New Zealand are encouraged[38] by the GCSB to select IT security products from this list. A similar list, known as the *Evaluated Products List* (EPL) operated by DSD in Australia. Commercial organisations can gain a competitive advantage by having their products listed on the PPL or EPL, especially when trying to penetrate government markets.

Table 6-1 AISEP certified, and in-evaluation firewall products, to February 1998.

Manufacturer	Description	Status	Assurance Level
CyberGuard Corporation	**CyberGuard Firewall**, version 2.2.1e	UK ITSEC Certificate 97/78	E3
Sun Microsystems	**SunScreen SPF-100G**, version 1.0	Certificate 96/01, December 1996. (No longer Available)	E1
Check Point Software Technologies Ltd.	**Check Point FireWall-1**, version 3.0 GOV (Special Evaluated Government Version)	*in-evaluation*	E3
Cisco Systems	**Cisco PIX Firewall**, version 3.0	*in-evaluation*	E1
IBM	**IBM Firewall**, version 3.1.1 — for AIX and Windows NT	*in-evaluation*	E3
Norman Data Defense Systems (Development) Inc.	**Norman Firewall**, version 4.0	*in-evaluation*	E3
Softway Pty Ltd, Trusted Information Systems Inc.	**Secure-IT Gauntlet**, version 3.2	*in-evaluation*	E3
Sun Microsystems	**SunScreen SPF-100**, version 1.0 — with Patch 102946-07	*in-evaluation*	E1

[38] Selection of products from the PPL is not mandatory for Government departments. However, if a product which meets the departments requirements, and budget, is listed on the PPL it is usual for it to be chosen over a similar unlisted product.

Table 6-2 UK ITSEC Scheme certified, and in-evaluation firewall products, to February 1998.

Manufacturer	Description	Status	Assurance Level
CyberGuard Europe Ltd.	CyberGuard Firewall, version 2.2.1e	Certified March 1997 (UK)	E3
Trusted Information Systems (UK) Ltd.	Gauntlet Internet Firewall, version 3.2	Certified August 1997 (UK)	E3
Milkyway Networks Corporation	Black Hole (SecurIT) Firewall, version 3.01E2	Certified August 1997 (Canada)	E2 / EAL3 (CC)
Check Point Software Technologies Ltd.	Check Point FireWall-1, version 3.0	*in-evaluation (UK)*	E3
The Knowledge Group	VCS Firewall	*in-evaluation (UK)*	EAL1 (CC)

6.4 Commercial Certification

The only commercial organisation currently performing firewall certification through product testing is the *International Computer Security Association* (ICSA) – previously known as the National Computer Security Association. The ICSA is based in the US, at Carlisle, Pennsylvania.

ICSA has a number of certification programmes[39] for a range of security products in addition to firewalls, including:

- Anti-Virus
- Cryptography
- Filtering and Monitoring
- Biometrics

The ICSA also provide certification for systems, such as WWW and Internet sites.

6.4.1 ICSA Firewall Certification

The ICSA believes that the fundamental motivation for a company to get its firewall product certified is to reduce both real and perceived risk. Customers and users of an ICSA certified firewall can be assured that they have taken due care in meeting minimum security standards that will protect them against common and well known attacks. This is perhaps more important in the US, than New Zealand, where it may decrease liability in the event of a security breach or failure. The concept of taking due care and reducing liability is also present in the CC EAL1 assurance level (see Section 6.3.2).

ICSA certification allows the certified firewall to point to a recognised security baseline, which they can show is met, or exceeded. It is possible that the use of a certified firewall could reduce insurance premiums, in the same manner as burglar alarms and dead-locks. It may even be possible to insure a network against damage done by an attacker.

[39] A full description of the development of generic ICSA evaluation criteria is available at **http://www.ncsa.com/services/certification/about.htm**

Commercial, marketing, and other competitive forces, provide the impetus for vendors and manufacturers to have firewall products certified. It is highly probable that a vendor would seek product certification because competing products have been certified. This trend can be seen in the leading firewall manufacturers which have all attained an ICSA firewall certification (see Table 6-3). The same group of manufactures are also having their firewalls evaluated through either the AISEP (see Table 6-1), or the UK ITSEC Scheme (see Table 6-2).

Table 6-3 ICSA certified firewalls, to March 1998.

Manufacturer	Operating System	Description	Status
3Com Corporation	Proprietary	NETBuilder 9.1	in-testing
ANS Communications, Inc.	Solaris	ANS Interlock	in-testing
Ascend Communications, Inc.	Proprietary	Pipeline Router Plus	in-testing
Bull S.A	AIX	NetWall	certified
CheckPoint Software Technologies, Inc.	Solaris	CheckPoint Firewall-1	certified
	Windows NT	CheckPoint Firewall-1	certified
Cisco Systems, Inc.	Proprietary	Private Internet Exchange	in-testing
	Windows NT	Centri Firewall	certified
Cyberguard Corporation	UnixWare	CyberGuard Firewall	in-testing
Digital Equipment Corporation	DEC UNIX	Digital Alta Vista Firewall '97	certified
	Windows NT	Digital Firewall '97 NT	certified
Elron Software, Inc.	MS-DOS	Elron Firewall	certified
Global Technologies Associates, Inc.	BSDI	GFX Internet Firewall System	in-testing
	Proprietary	GNAT Box	certified
IBM	AIX	AIX	certified
	OS 400	Firewall for AS/400	in-testing
Internet Devices	Solaris	AFS 2000	certified
Internet Dynamics	Windows NT	Conclave	certified
Isolation Systems Limited	Proprietary	InfoCrypt Enterprise Version	certified
Livermore Software Laboratories, Intl.	AIX	PORTUS	certified
Lucent Technologies, Inc.	Proprietary	Lucent Managed Firewall	certified
Milkyway Networks, Inc.	Sun OS	Black Hole	certified
NetuGuard, Ltd.	Windows NT	Guardian	in-testing
Network-1 Software & Technology	MS-DOS	Firewall/Plus	in-testing
RadGuard, Ltd.	Proprietary	Pyrowall	certified
	Proprietary	Crypto Wall	certified
	Proprietary	clPro-fw	certified
Raptor Systems, Inc.	Sun OS	Eagle	certified
	Windows NT	Eagle NT	in-testing
	HP-UX	Eagle	certified
Secure Computing	BSD	Sidewinder	certified
	BDS "Janus"	BORDERWare	certified
	Windows NT	NT Firewall	certified
Sun Microsystems, Inc.	Proprietary	SunScreen SPF-200	certified
	Proprietary	SunScreen EFS	certified
Technologic, Inc.	BSDI	Interceptor Internet Firewall	certified
Trusted Information Systems, Inc.	BSDI	Gauntlet Internet Firewall	certified
Ukiah Software, Inc.	Novell IntranetWare	Netroad Firewall	in-testing
WatchGuard Technologies, Inc.	Windows NT	Watchguard Security Management	certified

The FWPDC[40] (see Appendix B) unlike the ITSEC and CC has no notion of assurance levels, instead the result of applying the criteria to a firewall product is either a *pass* or *fail*. The FWPDC is focused on testing the resistance of firewalls against a standard set of vulnerabilities, as opposed to the ITSEC and CC which assess the fundamental design and engineering principles of the underlying technology. In effect the FWPD Criteria takes a "black-box" approach to evaluating firewall products.

One of the problems with firewall certification under government schemes such as the AISEP and UK ITSEC Scheme is the length of time required to complete an evaluation. The ICSA argues that the digital world moves far too quickly to certify only a particular version of a product or incarnation of a system. Therefore, the various ICSA certification criteria and processes are designed so that once a product or system is certified, all future versions of the product (or updates of the system) are inherently certified. This is normally accomplished by three means:

1. A contractual agreement is made between the ICSA and the product vendor or the organisation that owns or runs the certified system, agreeing that the product or system will be maintained at the current, published ICSA certification standards. It is expected that the organisation's own quality assurance programs will incorporate the current ICSA certification criteria as a part of their continuous product or system development processes. This means that a significant part of the ICSA certification process involves self-checking by the organisation whose product or system is certified.

2. The ICSA or its authorised agents normally perform random spot checking of the current product (or system) against current ICSA criteria for that certification category. Products or systems are typically spot-checked for current compliance two to four times each year. If a product or system fails a spot check, the responsible party is given a short time (typically 2 to 4 weeks) to rectify the problem(s). If the shipping product or production system still does not meet current certification criteria by the end of this grace period, then ICSA certification is explicitly and publicly revoked.

3. ICSA certification is renewed annually. At renewal time, the full certification process is usually repeated for the current production system or shipping product against the current criteria.

These steps ensure that ICSA certification is relatively independent of product or system updates and version changes, therefore owners and users can be assured that the current version of the product or system meets the current ICSA certification criteria.

6.4.2 ICSA Firewall Testing

The FWPDC reduces the problem of firewall evaluation by considering only a subset of possible vulnerabilities. The ICSA commissions or performs studies, surveys and other research to ascertain the relevant risks. Those which are exceedingly rare, merely theoretical, and of trivial impact are discarded. In order to keep up with constantly evolving risks the FWPDC is updated on a regular basis from the results of the ICSA's ongoing research. The benefit of this is that the criteria, and more importantly the testing regimes, continue to address the latest vulnerabilities introduced by the rapid evolution of firewall technology.

The FWPDC, and related testing standards are developed by experts, with input from third party groups. Actual evaluation of firewall products and systems is normally implemented by skilled evaluators under expert supervision and oversight. Evaluation is performed either by ICSA personnel and labs or by third-party labs and personnel trained and authorised by ICSA for this purpose. Evaluators are trained through an internal ICSA process which consists of formal and elective components [Cafarchio, 1998].

[40] The current version of the FWPDC is available at **http://www.ncsa.com/services/consortia/firewalls/certification.htm**

As a design goal, testing is automated where possible, and checklist oriented when not. The test protocols are designed to be reproducible, objective and unambiguous. The ICSA believes that to be appropriate, and to meet the needs of the commercial sector, certification has to be inexpensive, and accomplished with a rapid turn around. This approach allows a firewall product evaluation to be completed typically within 2 weeks, although in some cases evaluation has taken longer than 6 weeks [Cafarchio, 1998]. Obviously a short evaluation time-scale is very attractive to sponsors because it does not significantly delay marketing of certified firewall products. Therefore, sponsors can begin to recover their investment in certification much sooner than government certification schemes would allow. In addition, the ICSA concept of continual certification helps to simplify the issue of product modification — as long as the firewall product meets the current FWPDC it continues to be certified. If a firewall fails a random certification check the sponsor is given 3 weeks to change the product and release a patch, otherwise the firewall is publicly decertified [Cafarchio, 1998].

The FWPDC outlines an evaluation based on penetration testing using the most current version of the following commercial testing tools:

- *ISS SAFEsuite – Internet Security Systems* (ISS) SAFEsuite is a family of security analysis tools, including Internet Scanner (vulnerability analysis, i.e. penetration testing), System Security Scanner (system assessment), and RealSecure (intrusion detection). Information about SAFEsuite is available at **http://www.iss.net/prod/products.html**

- *Ballista/CAPE* – Ballista is a network security auditing tool developed at Secure Networks Incorporated. CAPE (Custom Auditing Packet Engine) is a tool included with Ballista which can perform protocol level spoofing and attack simulations. Information about Ballista/CAPE is available at **http://www.secnet.com/navlb.html**

- *Kane Security Analyst – Kane Security Analyst* (KSA) is a product from Intrusion Detection Corporation and provides a security assessment for Windows NT and Novell operating systems. Information about KSA is available at **http://www.intrusion.com/product.htm**

- *NetSonar* – Netsonar is a product of the WheelGroup Corporation and is a network vulnerability analysis and mapping tool. Information about NetSonar is available at **http://www.wheelgroup.com/netsonar/sonar.html**

In addition to these commercial tools, other software tools such as port scanners, and custom written exploitation scripts are used to carry out the penetration testing and vulnerability analysis.

The FWPDC is suitable for evaluating firewall products that will operate in low to medium threat environments where attackers have limited resources and ability. In high threat environments attackers may have access to tools and techniques that are outside the scope of the penetration tests conducted as part of the FWPDC evaluation.

6.5 Summary

A certified firewall product has the benefit of an independent assessment using a criteria which allows the award of a distinct level of assurance to the firewall's security claims and objectives. When considering a specific installation the value of the information and systems protected by the firewall, and the threats to them, need to be considered. Often the more valuable the information and systems the higher the threat. If these threats can be countered by the features or assurance of a trusted product, then it is certainly worthwhile to consider them in a purchase decision.

In general corporate and government environments where information and systems are unclassified, the assurance provided by the FWPDC, the ITSEC (E1 to E2) and CC (EAL1 to EAL3), is sufficient to indicate that a firewall will withstand all of the well known automated attacks launched by tools such as ISS and SATAN. Therefore ICSA and low assurance ITSEC and CC based certification is most suitable in low to medium threat environments.

For systems that process highly sensitive corporate, or unclassified-but-sensitive (UBS) government information, it is arguable whether these systems should be protected by FWPDC or low assurance ITSEC and CC evaluated firewalls. At these assurance levels the criteria used are focused on penetration testing, they do not consider in enough detail (if at all) aspects such as, user and administrative documentation, delivery or manufacturing processes, source code, or covert channels. Therefore, in medium to high threat environments where the compromise of systems and information would have a serious impact on an organisations credibility, competitiveness or survival, only firewalls certified at and above ITSEC E3 and CC EAL4 assurance levels should be considered. It is only at these and greater assurance levels that sufficient consideration is given to the firewall's fundamental engineering and design processes.

Initially, IT security evaluation was carried out by a government security agency such as New Zealand's GCSB, the United Kingdom's CESG, or the US's NSA. Unfortunately, this is not necessarily an ideal environment in which to carry out a wide range, in-depth, security evaluation. Frequently the focus and needs of the government and related bodies differ from the needs of industry in general. Further, procedures typically move very slowly in such organisations and therefore firewall manufacturers and vendors may refrain from undertaking a longer government certification in favour of obtaining a certificate through a shorter commercial certification process. The major benefit of such certification procedures is that typically penetration testing can be achieved in a relatively short period of time by using more simplistic criteria.

It is widely accepted that for nationally classified military or government systems, current firewall technology is not sufficiently advanced to protect connections to untrusted networks, such as the Internet. The only solution under these circumstances is to retain the physically separated classified network, and provide stand-alone Internet connections or separate unclassified networks. Evaluations are typically carried out to assurance levels of up to E3 or EAL3 today. Until high assurance mechanisms that can control the cross-over of information between classified and unclassified networks are available, then there will always be a need for disparate networks.

Chapter 7. Virtual Private Networks

7.1 What is a Virtual Private Network?

A *Virtual Private Network* (VPN) is a conceptual network formed by defining a closed group of users and encrypting all communication between its members. It is important to understand what the terms "virtual" and "private" mean when used to describe a VPN. By definition a VPN does not exist unless it is both virtual and private:

- *Virtual* – a VPN is *virtual* in that the connections formed are not part of a dedicated network of traditional network infrastructures (e.g. *Digital Data Network* (DDN), *Integrated Services Digital Network* (ISDN), *Packet Switched Network* (PSN), *Packet Switched Telephone Network* (PSTN), and *Frame Relay*) from a telecommunications provider. Instead, connections are logically partitioned and transmitted over public data networks such as the Internet.

- *Private* – a VPN is *private* (or confidential) because the traffic it handles is encrypted while it travels between the end points of the VPN. In the case of the Internet the end points would normally be located at each organisation.

A precursor to the VPN was the *closed user group* (CUG) defined as part of the CCITT[41] X.25 packet switched network standard [CCITT, 1988]. The CUG is an optional facility which allows nodes to form a group to which access restrictions can be applied. Although unauthorised connection to nodes within or outside the CUG can be prevented, there is no provision for confidentiality. With a protocol analyser an attacker with access to the intervening network can capture and view the data sent by any CUG node.

The idea of privacy in relation to network ownership should not be confused with privacy in relation to confidentiality. For example, traditional networks owned and operated by a telecommunications provider can be considered private because they are not generally accessible by the public. However, an attacker gaining unauthorised access to the "private" network will still be able to examine data transmitted across it. Currently, both Clear Communications and Telecom market such "VPN" solutions. However, in both cases the VPN service is simply a guarantee that an organisations network traffic will stay within the network architecture owned and operated by the telecommunications provider — this is simply privacy through obscurity!

VPNs are growing in importance as organisations look for cost effective strategies for connecting geographically dispersed users and networks. Traditional WANs have typically been very expensive, particularly when alternative routing is installed for resiliency. A number of public claims have been made regarding the savings achieved by moving network traffic from leased lines to the Internet. For example DMW Worldwide (Colorado Springs, US) have reported that replacing their dedicated leased lines with encrypted frame relay has reduced expenses by US$60,000. In another case, Phillips Tarifica (London, UK) reported that replacing a 64 Kbit.sec^{-1} leased line between London and Tokyo which cost US$159,174 per-year to operate with an Internet-based VPN connection reduced this figure to US$20,000 [Cray, 1997]. While it is widely accepted that savings can be made, not every leased line can be suitably replaced by an Internet based VPN. This is due to a number factors:

- *Quality of Service* (QoS) – IPv4 does not provide QoS, e.g. a transport delay of less than 100 msec cannot be guaranteed for a live video conference across the Internet.

- *Bandwidth* – is an issue of QoS but has a direct implication for moving traditional network infrastructures to the Internet. For example an ISP cannot guarantee an organisation a constant

[41] CCITT is an acronym for the International Telegraph and Telephone Consultative Committee which is the data and telecommunications standards body of the International Telecommunications Union (ITU) which, like the International Standards Organisation (ISO), is an agency of the United Nations.

512 Kbit.sec^{-1} connection across the Internet because bandwidth is distributed dynamically between all users.

- *Standards* – there are many open and proprietary VPN standards, however interoperability is a problem in areas such as key management, cryptographic protocols, and vendor specific options.

Using the Internet as a network bearer can provide organisations with a low cost alternative to traditional networks. Connection to the Internet allows an organisation to use the same network to connect local and remote locations, while at the same time providing employees with access to Internet-based applications such as email and the WWW. In addition the organisation benefits from the inherent resilience of the Internet — not surprising considering that ARPANET was originally designed to withstand nuclear attacks!

Unfortunately, since the Internet consists of many interconnected public data networks the confidentiality, integrity, and availability of any data transported across it cannot be assured. This is a problem for any organisation that wishes to transmit sensitive information, and requires guaranteed access to their systems and networks. Desire for cost effective communications and data security has driven much of the development of Internet-based VPN technology. Although the focus of this Chapter is on establishing VPNs across the TCP/IP networks such as the Internet the concepts are equally applicable to most network protocols, and can be implemented regardless of network topology and size (i.e. LAN, MAN, or WAN).

7.2 Virtual Private Networks and the Internet

Increasing numbers of corporate, government, and academic organisations are realising that integrating secure data transmissions within the framework of the Internet reduces their costs and expands their capabilities. For example, a VPN could allow a travelling salesperson to access sensitive customer records held on a database at the corporate headquarters — applications such as this can lead to significant competitive advantage. Internet-based VPN solutions are normally provided in firewalls, routers, or standalone encryption devices. With functionality ranging from encrypting everything to being application or protocol specific.

VPNs are particularly useful for creating secure *Extranets*. An Extranet represents a collaborative network established by organisations that share common goals and have a requirement to exchange information. Instead of using traditional network architectures an Extranet is implemented using the Internet and related technologies such as TCP/IP, WWW, Java, ActiveX, etc. Extranets are typically used to link businesses with their suppliers and customers, often in an effort to reduce paperwork and speed up order placement/payment. Some examples of Extranet applications include:

- The use of newsgroups or groupware by co-operating organisations to share knowledge and experience.
- Training programs or other educational material developed and shared between organisations.
- Shared product catalogues accessible only to wholesalers or re-sellers.
- Support management and control for organisations that are part of a common project.

Obviously such applications may be considered sensitive by the organisations involved, thus a VPN can be used to ensure strong authentication and protect the confidentiality of sensitive information exchanges.

Intranets which are built using the same Internet technologies as Extranets can also benefit from the application of VPN technology. In essence an Intranet is a localised version of the Internet operating within an organisation's own networks. Intranets are generally implemented to enable the sharing of information and computing resources throughout an organisation. In many cases the roles of employees

dictate the information and computing resources to which they require access. For example, employees in the Marketing department will normally have no requirement for access to the Account department's payroll database. In such a case a VPN could be used to authenticate access requests to the payroll database, and protect the confidentiality of payroll information as it transits the organisation's networks.

The last major area in which VPN technology is being applied is in support of the teleworker[42] (or telecommuter). A teleworker is an employee that works outside the traditional office or workplace, usually at home or in a mobile situation (e.g. an employee on a business trip, or a travelling salesperson). According to one study, teleworking has been growing at 15% a year since 1990 in the US, while 80% of Fortune 1000 companies are likely to introduce it within the next two to three years. In contrast, a New Zealand survey of 500 New Zealand companies reported that from 314 responses 38% were considering teleworking while 51% were not [Computerworld, 1998]. Although work at the organisation's premises is unlikely to disappear, new forms of telecommunication such as video conferencing and groupware are likely to make teleworking more prevalent in the future.

Figure 7-1 illustrates a number of ways in which VPNs can be deployed to protect sensitive information transfers. Note the use of two Secure IQ VPNet[43] routers to create a VPN between Extranet partners, and teleworker. The Secure IQ VPNet routers encrypt all data communicated between Organisation A and B, but also allow unencrypted traffic to flow.

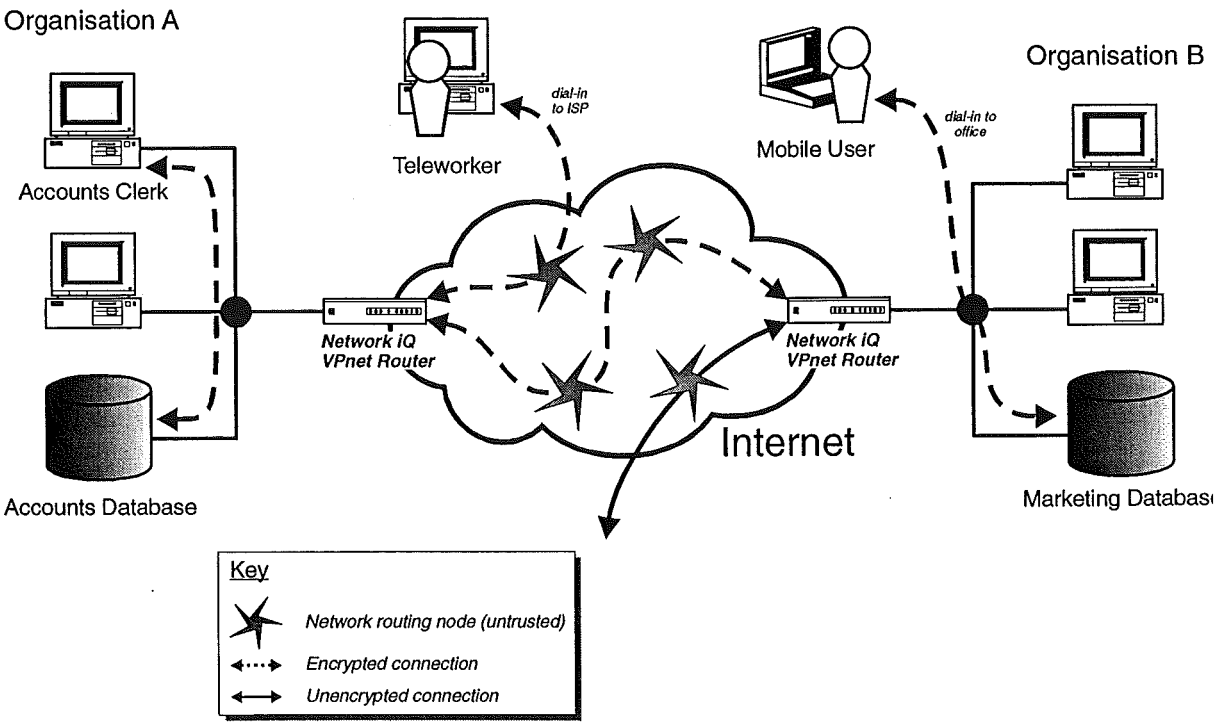

Figure 7-1 Extranet secured by a VPN.

The basis of all VPNs is cryptography — the art and science of keeping messages secure. Section 7.3 provides a basic overview of the cryptographic concepts used in the Sections that follow.

[42] A very useful and interesting WWW-site which contains a great deal of information and links to telecommuter resources is available at **http://www.mother.com/~dfleming/dmflinks.htm**

[43] Secure IQ VPNet routers are products of Teltrend Inc. (Chicago, US). Teltrend's router products are designed in Europe and New Zealand. Information about Teltrend's New Zealand operation can be found at **http://www.securicor.co.nz**

7.3 Overview of Cryptography

Cryptography is the art and science of keeping messages secure. It is the process of changing plaintext into encrypted, or ciphertext messages. The opposite process is employed by the receiver where the ciphertext is decrypted back to the original plaintext message. Enciphering and deciphering a message is achieved by applying a mathematical function known as a *cryptographic algorithm.* The most obvious reason for using cryptography is to protect the confidentiality of messages. However, it is also used to achieve a number of complimentary security objectives:

- *Authentication* – enables the message receiver to ascertain the messages origin; preventing an attacker masquerading as a legitimate participant.
- *Integrity* – the receiver is able to verify that the message has not been modified in transit; an attacker cannot substitute nor alter the original message.
- *Nonrepudiation* – prevents the sender from denying they sent the message.

The majority of cryptographic algorithms used on the Internet, and form the basis of most VPNs, have been placed in the public domain — here they are then open to a wide range of scrutiny and cryptanalysis[44]. Proprietary (or restricted) algorithms are commonly found in low-security commercial applications and tend to be inherently insecure [Schneier, 1996]. Although high-grade military algorithms are classified, this is done to increase the difficulty of cryptanalysis — their security like those found in the public domain is based on a key, or keys. Keys are used as input variables to make the outcome of the algorithm unique. The key is generally chosen from a vast number of values — the total number of which is known as the *keyspace.* Generally the larger the keyspace the harder it is to break the ciphertext through brute force attacks which involve trying every possible key.

There are two algorithmic approaches generally used to secure Internet based communications. The first, known as *symmetric or secret-key*, uses a single key to both encrypt and decrypt a message. The second type use a different key for the encryption and decryption processes — algorithms of this nature are known as *asymmetric or public-key*. Both of these approaches discussed in the following Sections.

7.3.1 Secret-Key (Symmetric) Cryptography

As secret-key algorithms use the same key for both encryption and decryption it is necessary for both the sender and receiver to share a copy of the key. Since the same key is used by both parties to encrypt and decrypt messages sent to one another it has to be kept secret. This presents a number of problems, the least of which is "how to distribute the key to all participants without it being compromised?" In military and intelligence organisations this is usually achieved through "safehand" courier[45]. For commercial organisations the cost of safehand delivery may be prohibitive — instead a simple telephone call, or registered mail may be employed. It is not enough to simply distribute a key securely, it must also be stored in a secure manner — this is often achieved physically by using a safe, or increasingly by storing the key on an electronic/magnetic storage device protected with a *personal identification number* (PIN). Another problem occurs in the management of the keys, especially when many participants are involved or when different keys must be used by different groups. The problem starts to become insurmountable when parties at either end do not know each other — or a secure transmission channel does not exist.

[44] Cryptanalysis is the art and science of breaking ciphertext.

[45] The term "safehand" is used to describe the process by which an item is physically transported between destinations by a trusted person (or courier). The courier remains in contact with the item at all times until it has been delivered.

Symmetric algorithms are often represented by the following mathematical functions:

$$E_K(M) = C \qquad \textit{encryption function}$$
$$D_K(C) = M \qquad \textit{decryption function}$$

Key

K	*encryption/decryption key*
M	*plaintext*
C	*ciphertext*

These functions have the property that (see Figure 7-2):

$$D_K(E_K(M)) = M$$

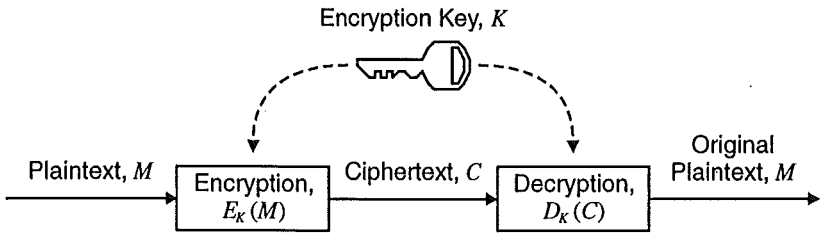

Figure 7-2 Symmetric algorithm encryption and decryption.

Symmetric algorithms can be categorised as either *stream algorithms*, known as *stream-ciphers*, or *block algorithms*, known as *block-ciphers*. Stream-ciphers operate on the plaintext one bit (or byte) at a time. An example of a stream algorithm is the Rivest Cipher 4 (RC4) developed in 1987 by Ronald Rivest for RSA Data Security Inc. It remained a trade secret until September 1994 when the source code was anonymously posted to the Cypherpunks mailing list. RC4 can be exported from the US if its key length is 40-bits or less. It has been implemented in many commercial products, including Lotus Notes, Oracle Secure SQL, and is part of the Cellular Digital Packet Data specification [Schneier, 1996].

Block-ciphers operate on groups of bits known as blocks — a block size of 64-bits is typical. The Data Encryption Standard (DES) is a block algorithm and perhaps the most well known cryptographic algorithm. DES has been a world wide standard since 1976 when it was adopted as a US Federal standard for use on all unclassified government communication. It was originally developed by IBM during the 1970s and was known then as *Lucifer*. Upon review by NSA, alterations were made and the key size was reduced from 128-bits to the current 56-bits. DES is still a popular algorithm and can be implemented in both hardware and software. Financial institutions have a large installed base of DES encryptors used to protect their WAN communications. However, it is felt by many that DES in its original form may be coming to the end of its life. The reason is that modern cryptanalysis and increasing computer power have significantly reduced the strength of 56-bit DES, particularly to brute force attacks. However, the latest 56-bit DES cracking competition (launched on January 13, 1998, and known as "DES Challenge II") sponsored by RSA Data Security, Inc., still required 22,000 participants linking together over 50,000 CPUs and 39 days to search 85% of the key space to recover the plaintext message — "Many hands make light work" [RSA, 1998].

The greatest advantage of using secret-key cryptography is that it tends to be very fast, even on slow computers. This is because the algorithms are generally small and perform simple computations that can be optimised in either hardware or software.

7.3.2 Public-Key (Asymmetric) Cryptography

In contrast to secret-key algorithms, public-key algorithms are designed so that the encryption key is different form the decryption key. Both keys are mathematically dependent upon one another — messages encrypted by the one key, can only be decrypted by the other key, and vice versa. An important requirement for asymmetric cryptography is that the decryption-key must be computationally unfeasible to compute from the encryption-key. The algorithms are known as "public-key" because the encryption-key can be made public. Anyone may use the public-key to encrypt a message but only the holder of the corresponding decryption key can decrypt the message. The decryption key is also known as the *private-key*.

An alternative and increasingly important use of public-key cryptography is its ability to digitally sign messages. This is analogous to a hand-written signature on a message or document, providing proof of authorship. Digital signatures are discussed further in Section 7.3.3.

Asymmetric algorithms are often represented by the following mathematical functions:

$$E_{K_1}(M) = C \qquad \textit{encryption function}$$
$$D_{K_2}(C) = M \qquad \textit{decryption function}$$

Key

K_1	*encryption key*
K_2	*decryption key*
M	*plaintext*
C	*ciphertext*

These functions have the property that (see Figure 7-3):

$$D_{K_2}(E_{K_1}(M)) = M$$

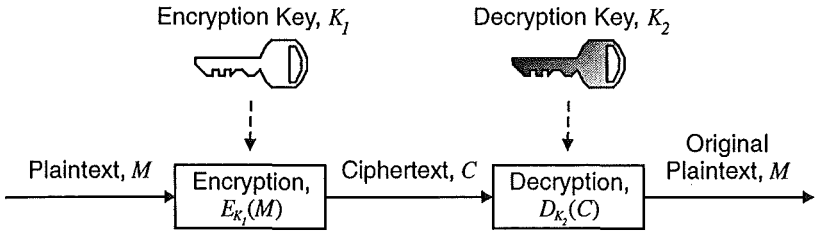

Figure 7-3 Asymmetric algorithm encryption and decryption.

In general, Person A retrieves Person B's public-key from a public-key server and uses this to encrypt their message to Person B. Since Person B's private-key is the only one which can decrypt the message, Person A can be certain that only Person B can read it (see Figure 7-4). Unfortunately, since everyone has access to Person B's public-key, although he can read it, he cannot be certain of the authenticity of Person A. The exchange is private but not authenticated.

This is why a trusted and authenticated key distribution system is necessary to support the use of public-keys on the Internet — one such system known as a *Certification Authority* (CA) is discussed further in Section 7.3.4. Another disadvantage of public-key algorithms is that they require much greater computational power — in general they are 1000 times slower than symmetric algorithms [Schneier, 1996].

In most practical applications public-key cryptography is not a substitute for secret-key cryptography. Public-key algorithms are two slow, and are vulnerable to chosen plaintext attacks [Schneier, 1996]. As a result, a *hybrid cryptosystem* is most often employed. Such a system uses public-key cryptography to

securely distribute *session keys* for use with symmetric algorithms. The benefit of this approach is that the computational expensive asymmetric algorithm need only be used once to encrypt a relatively short random secret-key. The secret-key is then used with a computationally less expensive symmetric algorithm to secure the subsequent communications.

Hybrid cryptosystems negotiate the session-key at the beginning of a session; on completion the keys are securely deleted. This is referred to as the *Diffie-Hellman* (DH) technique, named after its inventors Whitfield Diffie and Martin Hellman. Invented in 1976, the DH key exchange algorithm was the first openly published example of public-key cryptography. It is based on the difficulty of calculating discrete logarithms in a finite field, as compared with the ease of calculating exponentiation in the same field [Schneier, 1996]. The advantage of DH key exchange is that unknown parties can negotiate secret-keys on the fly, according to their session needs.

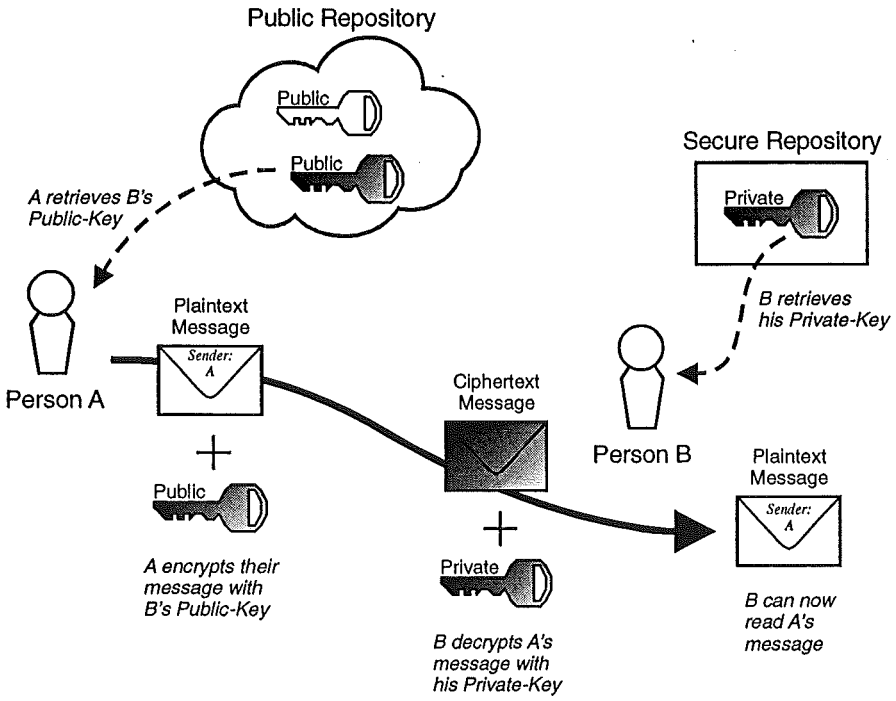

Figure 7-4 A secure message exchange using public-key cryptography.

The best known and most popular public-key system is RSA, named after Ronald Rivest, Adi Shamir, and Leonard Adleman who developed the algorithm while at MIT. RSA gets its security from the difficulty of factoring large numbers. The public and private-keys are functions of a pair of large prime numbers — commonly between 100 and 200 digits.

Public-key cryptography is the most flexible, scaleable, and efficient way for users to obtain the shared secrets and session keys needed to support the large number of ways Internet users can securely interoperate. The great advantage of the public-key cryptography over secret-key cryptography is that, theoretically, no confidential information need be exchanged between participants before secure communication is possible. A person requires only the public-key of the party to whom they are sending a message. The problem is knowing whether the public-key is authentic. Public-key servers can be impersonated and public-keys can expire. If an attacker is impersonating the public-key server they can replace the recipients public-key with their own. This allows the attacker to act as a man-in-the-middle, viewing and relaying all of the messages sent by the intended participants.

7.3.3 Digital Signatures

A number of public-key algorithms can be used to generate digital signatures — RSA is such an example. In the case of a suitable public-key algorithm the digital signature is created by encrypting the message with the private-key. The recipient can then check the signature by decrypting the message with the senders public-key.

In most practical situations it is too inefficient to sign large messages using public-key algorithms — they are simply too slow. Instead digital signature protocols are implemented using one-way hash functions, referred to as either *secure hash functions* or *message digests*. The most common secure hash functions are *Message Digest 2* (MD2), *Message Digest 5* (MD5), and the *Secure Hash Algorithm* (SHA-1). Both MD2 and MD5 were designed by Ron Rivest, and are used in the *Privacy Enhanced Mail* (PEM) standard adopted by the *Internet Architecture Board* (IAB). SHA-1 was designed by the US *National Institute of Standards and Technology* (NIST) and NSA for use in with the *Secure Hash Standard* (SHS) [NIST, 1995]. SHS is applicable to all US Federal departments and agencies for the protection of unclassified information. It is also required for use with the *Digital Signature Algorithm* (DSA) as specified in the *Digital Signature Standard* (DSS), and whenever a secure hash algorithm is required for Federal applications.

A hash function works by taking a variable-length input string and returning a fixed-length string, known as a *hash value*. One-way hash functions ensure that it is hard to generate an input string which hashes to a particular value. This prevents an attacker from substituting a message with one having the same hash value — this is only possible if the message and digital signature are not encrypted before being sent. Other desired attributes of one-way hash functions include being collision free[46], output is independent of input, and a single bit change on average changes half of the bits in the hash value.

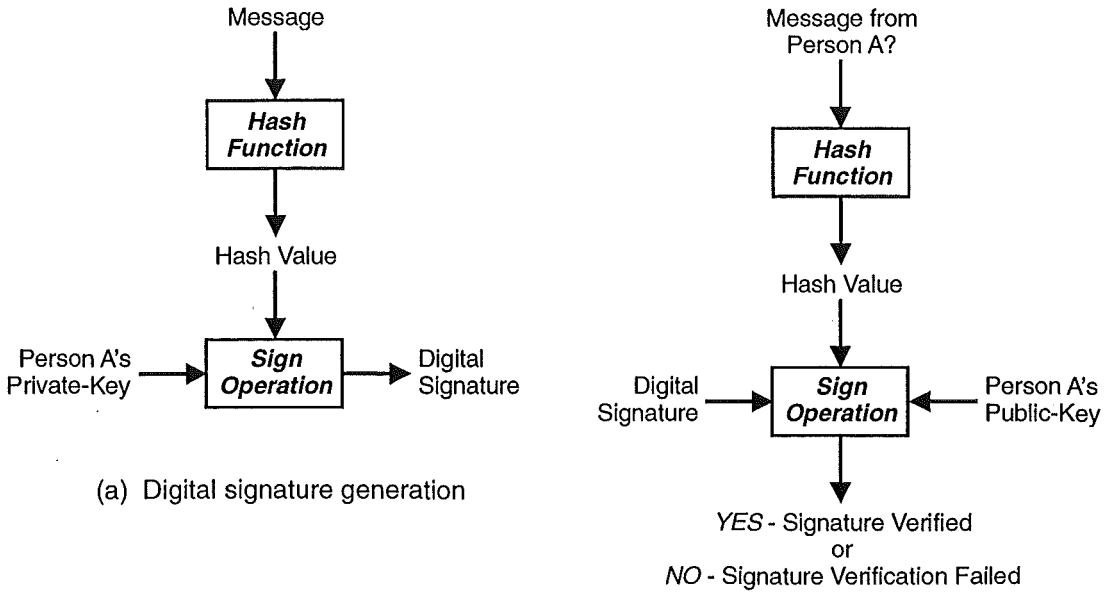

Figure 7-5 Digital signature generation and verification.

For example, instead of Person A signing their message they simply sign the hash value as shown in Figure 7-5a — this is far more efficient. The message and digital signature are then sent to the

[46] For a hash function which is collision free, it is hard to generate two input strings with the same hash value.

recipient(s). Of course this particular protocol requires that the participants agree on the digital signature algorithm and one-way hash function beforehand.

On receipt of the message and digital signature Person B computes a new hash value for the message and retrieves the original hash value by decrypting the digital signature using Person A's public-key. This verification process is shown in Figure 7-5b. If the new hash value is identical to the original then the message was signed using Person A's private-key — unless of course their private-key had been compromised

7.3.4 Certificate Authorities

Public-key and digital signature cryptographic systems both share a similar problem — the uncertainty of having the correct public-key. The combination of public-key and *third-party* signature is called a *certificate* — this term was first defined in 1978 by Loren Kohnfelder to refer to a signed record holding a persons common name and their public-key [Ellison et al., 1997]. The third-part which signs the certificate is commonly referred to as the *Certification Authority* (CA).

The most commonly used certificate structure is based on the ISO X.509 standard [X509, 1996] initially issued in 1988. X.509 is a collection of protocols used with the ISO authentication framework which provides authentication across networks. X.509 version 3 (or X.509v3) is the latest revision and addresses a number of deficiencies.

Figure 7-6 illustrates the major fields within the three versions of the X.509 certificate.

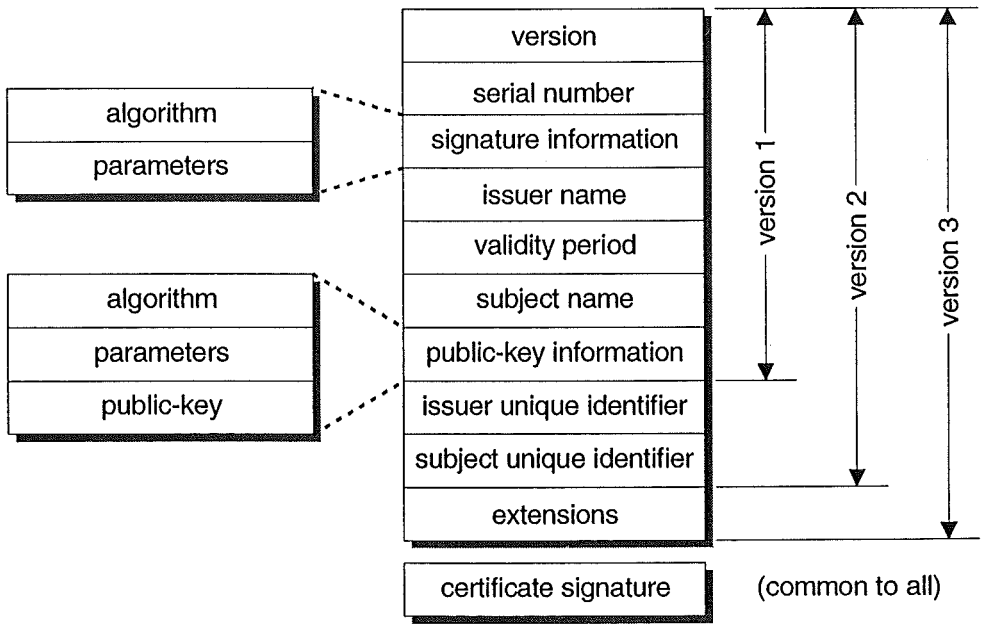

Figure 7-6 Three versions of the ISO X.509 certificate format.

The following describes the fields common to all versions of the X.509 certificate:

- *version* – identifies the certificate format (e.g. X.509v3).
- *serial number* – uniquely identifies the certificate within the CA that issued it.
- *signature information* – identifies the public-key algorithm used to sign the certificate (e.g. RSA) and any parameters it requires.

- *issuer name* – contains the name of the CA.
- *validity period* – contains a pair of dates between which the certificate is valid.
- *subject name* – contains the name of the user, or subject.
- *public-key information* – contains the subjects public-key, and identifies the public-key algorithm and any necessary parameters.
- *certificate signature* – this is the final field in all versions of the X.509 certificate. It contains the CA's signature of all the previous fields.

The version 2 certificate was a minor revision to version 1, and added the issuer and subject unique identifier fields for supporting directory access control. Version 3 enables extension fields to be added to the certificate. A number of predefined extensions exist, however anyone may define and add their own extension fields.

Unlike the certificate defined by Kohnfelder each user of an X.509 certificate is given a distinct name, or *distinguished name* (DN). As the DN is a combination of CA name and subject name it is unique across all CA's. This is necessary when searching for a subjects public-key certificate across multiple CA domains. Simply using the subjects common name could produce many ambiguous results.

Certificates are normally stored in a certificate database referred to as a *directory server* (DS). The DN is used to uniquely identify subjects certificates within the DS. Most commercial implementations of DSs (e.g. Netscape Directory Server) are accessed using the Lightweight Directory Access Protocol (LDAP) [Howes, 1995]. The alternative ISO X.500 *Directory Access Protocol* (DAP) requires implementation of an X.500 directory and the ISO protocol stack. The advantage of LDAP is that it interfaces directly to many standard databases and operates over the TCP/IP suite. It also requires less processing overhead and is significantly cheaper to implement than an X.500 based solution.

For example, if Person A wishes to communicate with Person B, Person A must retrieve Person B's certificate from the directory server. Person A then verifies the certificates authenticity. If both belong to the same CA, Person A simply verifies the CA's signature on Person B's certificate.

However, if Person A and B belong to separate CAs their certificates can only be verified by chaining backward and verifying the authenticity of each CA certificate until a common CA is found. The common CA certificate must then be authenticated by the CA which signed it — this effectively establishes a path of trust between Person A and B. Figure 7-7, page 87, illustrates the case where multiple CAs exist. A hierarchy of CAs functions similarly to the Domain Name Server system (DNS), in which a request for an IP address associated with a specific host name is passed up the DNS hierarchy until the name is resolved. In the case of a hierarchy of directory servers, each server in the system must have an authenticated relationship with the systems to which it is connected to be certain that the certificate received is authentic. If no common CA exists then a path of trust cannot be established between Person A and B — therefore it is not possible for them to verify each others certificate.

There are a growing number of CAs providing certificate services on the Internet. In fact most popular WWW-browsers, such as Microsoft IE4.0 and Netscape Communicator, come pre-configured with the certificates of common Internet CA's. For example, IE4.0 includes certificates for Microsoft, AT&T, Verisign, etc. These certificates can be used to verify that a WWW-site is authentic (i.e. they are who they say they are), or that content being downloaded (e.g. Java, ActiveX) can be trusted (i.e. the software is from the claimed author). Certificates are becoming an essential tool of Information Systems for building trust-relationships between users and organisations on the Internet.

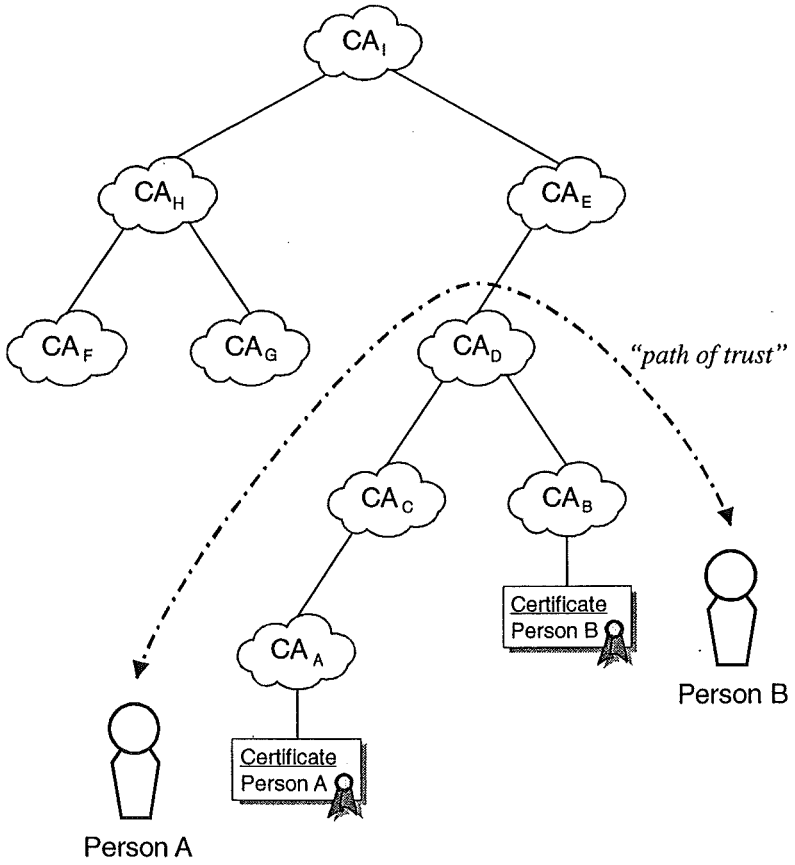

Figure 7-7 An example of a certification hierarchy.

7.4 Virtual Private Network Technology

The Internet allows organisations to interconnect their systems securely and cost effectively. To use the Internet as a VPN, the organisation's networks must be compatible with TCP/IP. Modern LANs often support a number of LAN-based protocols, including Novell's IPX, Apple's AppleTalk, and Microsoft's NetBEUI. These LAN-based protocols along with TCP/IP can operate side-by-side and simultaneously on a LAN, but are incompatible with the Internet. There are two issues to consider here, the first is providing private communications over the Internet, while the second deals with transporting incompatible LAN-based protocols over the Internet.

An organisation using TCP/IP internally can directly connect their private networks to the Internet. The only requirement being the use of officially-assigned Internet addresses. However, many organisations use unregistered IP addresses which cannot be used directly on the Internet. Sites most often solve this problem by using proxies or firewalls that perform IP masquerading or address translation (see Section 4.2). Thus an organisation needs only a single official IP address for, say, their firewalls external network interface — thus all traffic to the Internet appears to come from a single IP address. This is a much simpler and cheaper solution than registering an official address range and then changing the IP address of all the internal machines. It also has the added advantage of hiding the organisations internal network topology — knowledge of which can prove very useful to an attacker.

Organisations that wish to use the Internet to connect private networks using protocols other than TCP/IP must use some form of protocol *encapsulation* or *tunnelling*[47]. For example Microsoft's

[47] The terms "encapsulation" and "tunnelling" refer to techniques which enable network protocols to be transported over incompatible networks.

NetBEUI is incompatible with TCP/IP, however a NetBEUI packet can be encapsulated in an IP datagram allowing it to be sent across the Internet. At the destination, the IP datagram is decapsulated, i.e. the IP header is removed, leaving the original non-IP packet intact.

It is also possible to use an IP gateway to translate incompatible network protocols to IP. For example Novell NetWare clients running a WWW-browser on an IPX network can access Web servers on the Internet through Novell's IPX/SPX-to-IP Gateway with no modification.

Tunnelling is the best option for creating private IP-based VPNs. IP gateways introduce substantial overhead as they have to effectively convert one protocol stack to another. IP masquerading and conversion are useful for connecting private TCP/IP networks to the Internet, but do not address the problem of connecting non-IP private networks across the Internet.

Tunnels can either be *static* or *dynamic*. Static tunnels are created between sites that wish to remain connected for extended periods of time. Dynamic tunnels are suitable for session based connections such as WWW-browsing and are created on-demand whenever traffic is transferred.

Tunnelling protocols do not have to provide data security, many simply provide a way of transporting network protocols over incompatible networks. The most common tunnelling protocols are outlined below.

- *Generic Routing Encapsulation* (GRE) – specifies a protocol for performing encapsulation of an arbitrary Network-layer protocol over another arbitrary Network-layer protocol. GRE is specified in RFC 1701 [Hanks et al., 1994a]. The GRE protocol functions by encapsulating the Network-layer protocol to be transported in a GRE packet, which may optionally include route information. The resulting GRE packet can then be encapsulated in the final network protocol and delivered. RFC 1702 [Hanks et al., 1994b] is a companion memo which addresses the case of using IP as the delivery protocol or the payload protocol and the special case of IP as both the delivery and payload. GRE in itself does not provide encryption services.

- *Point-to-Point Tunnelling Protocol* (PPTP) – is a joint development by Ascend Communications and Microsoft. PPTP specifies a protocol which allows the Point-to-Point Protocol (PPP) to be tunnelled across an IP network. PPTP does not specify any changes to the PPP protocol but rather describes a new method for transporting PPP packets. PPTP supports tunnelling of IP, IPX, NetBIOS and NetBEUI protocols. The PPP packets are encapsulated using GRE, the resulting GRE packet is then delivered using an IP network. PPTP has been submitted to the *Internet Engineering Task Force*[48] (IETF) as an Internet-Draft [Hamzeh et al., 1997a]. PPTP provides proprietary cryptographic services to establish a VPN between a user's computer and the destination network (see Section 7.5).

- *Layer-2 Forwarding* (L2F) – focuses on providing a standards-based tunnelling mechanism for transporting Link-layer frames (for example, *High-Level Data Link Control* (HDLC), *asynchronous PPP,* or *PPP ISDN*) containing higher layer protocols [Cisco, 1997]. The L2F protocol is used to encapsulate the HDLC packet, the resulting L2F packet is then sent in a UDP datagram across an IP network. L2F supports tunnelling of IP, IPX, and AppleTalk protocols. In addition L2F allows the tunnel to be encrypted using IPSec.

- *Layer-2 Tunnelling Protocol* (L2TP) – is being designed by the IETF PPP working group and combines the best features from PPTP and L2F. L2TP is described in an IETF Internet-Draft

[48] The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The actual technical work of the IETF is done in its working groups (which are organised by topic into several areas, e.g. routing, transport, security, etc.), with much of the work being done through mailing lists. All technical reports, known as Internet-Drafts, produced by the working groups are published on the IETF WWW-site for public comment. An Internet-Draft expires after six months at which point it is updated (to incorporate necessary changes, comments, etc.), submitted for consideration as an RFC, or deleted (i.e. discontinued). Any interested individual can define and submit an Internet-Draft. The IETF WWW-site and all current Internet-Drafts are available from **http://www.ietf.org/**

document [Hamzeh et al., 1998]. No cryptographic services are defined in the L2TP standard, although IPSec could be used to secure the IP datagrams across an IP network.

- *Ascend Tunnel Management Protocol* (ATMP) – the ATMP protocol is currently being used in Ascend Communication products to allow dial-in client software to create a virtual presence on a user's home network from remote locations. The clients themselves are unaware of ATMP, although it is assumed that standard PPP or SLIP clients are being used [Showalter, 1996]. ATMP currently allows for both IP and IPX protocols to be tunnelled — encapsulation is performed using the GRE protocol. ATMP is defined in RFC 2107 [Hamzeh, 1997b]. It is interesting to note that Ascend Communications created PPTP's fundamental architecture and advanced the concept to Microsoft. The first implementation of PPTP was demonstrated at Networld+Interop in March 1995. No cryptographic services are defined in the ATMP standard, although IPSec could be used to secure the IP datagrams across an IP network.

- *Data Link Switching* (DLSw) – is a forwarding mechanism for the IBM SNA (Systems Network Architecture) and IBM NetBIOS (Network Basic Input Output Services) protocols. The protocol does not provide full routing, but instead provides switching at the SNA Data Link-layer (i.e. layer 2 in the SNA architecture) and encapsulation in TCP/IP for transport over the Internet. DLSw version 1.0 is defined in RFC 1795 [Wells et al., 1995]. RFC 2166 [Bryant et al., 1997] defines version 2.0 which is a set of backward compatible enhancements, the majority of which address scaling issues. No cryptographic services are defined in the DLSw standard, although IPSec could be used to secure the IP datagrams across an IP network.

- *Mobile IP* – is intended to enable nodes to move from one IP subnet to another. Mobile IP facilitates node movement from one Ethernet segment to another as well as accommodating node movement from an Ethernet segment to a wireless LAN. The principal design goal of Mobile IP is for a node to retain its IP address after it has moved to another network. The IETF Mobile IP working group has specified the use of encapsulation as a way to deliver datagrams from a mobile node's "home network" to an agent that can deliver datagrams locally by conventional means to the mobile node at its current location away from home [Perkins, 1996a]. Mobile IP specifies tunnelling for a number of circumstances such as firewall traversal. Other possible applications of encapsulation include multicasting, preferential billing, choice of routes with selected security attributes, and general policy routing [Perkins, 1996b]. IPSec is expected to be integrated with the Mobile IP implementation [Zao et al., 1997].

- *IP Security* (IPSec) – provides a security framework developed by the IETF IP Security Working Group for IP version 4 and IP version 6 [Atkinson, 1995a]. IPSec supports a number of tunnelling methods with or without encryption. A full discussion of IPSec follows in Section 7.6.

It is evident from the above descriptions of tunnelling standards that IPSec is becoming the predominant method for securing IP traffic. PPTP, ATMP, and L2F were primarily designed to connect remote dial-up users as virtual nodes to their home network — the remote user's computer appears as a physical node on the LAN. IPSec can secure the IP traffic between the ISP the remote user has dialled into and their home network. However, PPTP is the only tunnelling standard that can secure the dial-up connection from the user to their ISP for protocols other than TCP/IP.

Static and dynamic VPNs can also be created at the Application-layer of the TCP/IP suite. Application-layer VPNs are most commonly created using the *Secure HyperText Transport Protocol* (S-HTTP), *Secure Shell* (SSH), or *Secure Sockets Layer* (SSL). All three protocols are currently being developed as Internet-Drafts by various IETF working groups. The following points provide brief overviews for each protocol:

- *S-HHTP* – Secure-HTTP [Rescorla et al., 1997] is a secure message-oriented communications protocol designed for use in conjunction with HTTP. It is designed to coexist with HTTP's messaging model and to be easily integrated with HTTP applications. S-HTTP aware clients

can communicate with S-HTTP oblivious servers and vice-versa, although such transactions obviously would not use S-HTTP security features.

Interestingly, S-HTTP does not require client-side public-key certificates (or public-keys) because it supports symmetric-key only operation modes. This is significant because it allows secure transactions to occur without requiring users to have an established public-key. Although S-HTTP is able to take advantage of available certification infrastructures, its deployment does not require it.

S-HTTP supports end-to-end secure transactions, in contrast with the original HTTP authorisation mechanisms which require the client to attempt access and be denied before the security mechanism is employed. S-HTTP provides full flexibility in the choice of cryptographic algorithms, modes and parameters, and has been designed for extensibility. Option negotiation is used to allow clients and servers to agree on transaction modes (e.g., should the request be signed or encrypted or both?); cryptographic algorithms (e.g. RSA vs. DSA for signing, DES vs. RC2 for encrypting, etc.); and certificate selection. With S-HTTP, no sensitive data need ever be sent over the network in the clear.

- *SSH* – SSH[49] [Ylonen et al., 1997a] [Ylonen et al., 1997b] [Ylonen et al., 1997c] [Ylonen et al., 1997d] is a datagram-based binary protocol that is capable of functioning on top of any Transport-layer that can deliver a stream of binary data. It was originally designed as a replacement for the UNIX rlogin, rsh, and rcp commands, in addition, it is also used to provide secure X-Windows connections and secure forwarding of arbitrary TCP connections.

  SSH provides strong authentication and secure communications over unsecure channels. All communications are encrypted using IDEA or one of several other ciphers (e.g. triple-DES, DES, RC4-128, Blowfish). Encryption keys are exchanged using RSA, and data used in the key exchange is destroyed every hour (keys are never saved). Each host has an RSA key which is used to authenticate the host when RSA host authentication is used. Encryption is used to protect against IP-spoofing; public-key authentication is used to protect against DNS and route spoofing. RSA keys are also used to authenticate hosts. The datagram mechanism and related authentication, key exchange, encryption, and integrity mechanisms implement a Transport-layer security mechanism, which is then used to implement the secure connection functionality.

- *SSL* – SSL was initially designed by Netscape Communications, and is the predominant Application-layer security protocol. The SSL is a protocol layer which may be placed between a reliable connection-oriented Transport-layer protocol (e.g. TCP) and the Application-layer (e.g. HTTP). SSL provides for secure communication between a client and server by allowing mutual authentication, the use of digital signatures for integrity, and encryption for privacy. A discussion of SSL follows in Section 7.7.

The remainder of this Chapter discusses in detail the predominant VPN architectures used to protect dial-in connections (i.e. PPTP), the Network-layer (i.e. IPSec), and the Application-layer (i.e. SSL).

7.5 Point-to-Point Tunnelling Protocol

The PPTP is included with version 4 of the Windows NT Server/Workstation operating system, and is available as an add-on for the Windows 95 operating system. It is also supported in a number of *network access servers*[50] (NAS) available from companies such as Ascend Communications[51] (e.g.

[49] The following WWW-pages provide initial starting points for obtaining information about SSH; SSH homepage at **http://www.cs.hut.fi/ssh/** and SSH Frequently asked questions (FAQ) at **http://www.uni-karlsruhe.de/~ig25/ssh-faq/**

[50] Network access servers are also referred to as front-end processors (FEPs), dial-in servers or point-of-presence (POP) servers.

[51] Ascend Communications has a WWW-site at **http://www.ascend.com**

Ascend MAX 4000, 4002, and 4004), and 3Com[52] (e.g. Accessbuilder 8000). PPTP supports authenticated and confidential communication between a remote/mobile user and their home network.

Generally, there are three computers involved in a PPTP communication session:

- PPTP Client
- NAS
- PPTP Server

However, if a PPTP tunnel is created between a PPTP client and a PPTP server connected to the same LAN then a NAS is not required.

PPTP is commonly used to connect a remote PPTP client through a local ISP to a private enterprise LAN. A PPTP client makes two connections to establish a PPTP tunnel (see Figure 7-8, adapted from [Microsoft, 1997]). The remote client first establishes a dial-up, PPP based, network connection to the local ISP's NAS. The client then makes a second logical connection over the existing PPP connection to the enterprise PPTP server. The second connection creates a PPTP control connection, and the PPTP tunnel which consists of IP datagrams containing encrypted PPP packets. The control connection is used to establish, maintain, and end the PPTP tunnel. A full description of the PPTP protocol can be found in [Hamzeh et al., 1997a].

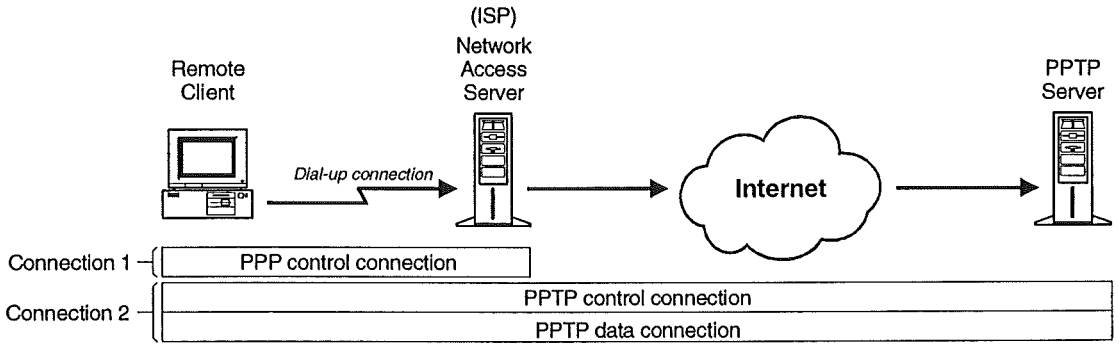

Figure 7-8 Creating a PPTP tunnel.

PPTP encapsulates the encrypted and compressed PPP frames into IP datagrams for transmission over the Internet. The IP datagrams are created using a modified version of the GRE protocol (see Section 7.4). These IP datagrams are routed over an IP network until they reach the PPTP server that is connected to both the IP network (e.g. Internet) and the private network. When the PPTP server receives the IP datagram from the IP network, it retrieves the original network packet (i.e. IPX, NetBEUI, or TCP/IP) sent by the remote client and delivers it across the private network to the destination computer. Retrieval of the network packet is achieved by decapsulating it from the PPP packet contained in the IP datagram and then decrypting it. This process allows the PPTP client to appear as a virtual node on the private network — that is, the PPTP client appears as if it is physically connected to the private network.

Figure 7-9, page 92, (adapted from [Microsoft, 1997]) illustrates the multi-protocol and encryption support of the PPTP. In addition Figure 7-9 incorporates protocol stack diagrams which indicates the protocol layers in use at each transport stage — these stack diagrams should be interpreted such that each higher level protocol encapsulates the one below it. Packets sent by the remote PPTP client to the PPTP server pass through the PPTP tunnel to their destination on the private network.

[52] 3Com Corporation has a WWW-site at **http://www.3com.com**

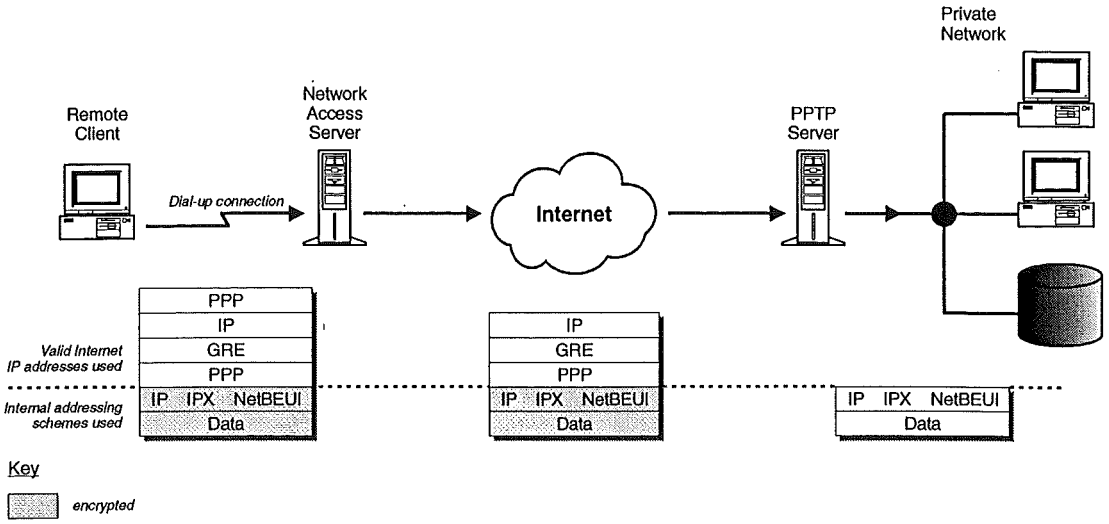

Figure 7-9 Connecting a remote dial-up PPTP client to the private network.

It is also possible for remote clients (e.g. UNIX, Apple Macintosh, Microsoft Windows 3.11) that are not PPTP-enabled to take advantage of PPTP by connecting to a PPTP-enabled NAS. In this case, the NAS instead of the remote client establishes the PPTP connection to a PPTP server. It is important to note that to establish a PPTP connection between an ISP's NAS and a private network the ISP would require sensitive information, such as the remote user's-id and password. Obviously providing identification and authentication information to a third-party raises serious security issues!

7.5.1 PPTP Security

PPTP provides authentication, access control, and cryptographic mechanisms to Windows NT Server/Workstation version 4.0, and Windows 95 PPTP clients. As PPTP traffic is carried in IP datagrams it is possible to use firewall technology to protect the PPTP server and private network. PPTP traffic uses TCP port 1723, and the IP datagram uses Protocol = 47 (see Section 2.3), as assigned by IANA. PPTP can be used with most firewalls and routers by enabling traffic destined for TCP port 1723 and Protocol = 47 to be routed through the firewall or router. For example, an organisation could use such packet filtering to block all traffic except PPTP ensuring that all network traffic entering and leaving their private networks is authenticated and encrypted.

The following points provide an overview of the authentication, access control, and encryption mechanisms implemented in the PPTP:

- *Authentication* – In order to use PPTP a remote client must first establish a connection to their ISP's NAS which may require the client to authenticate themselves. It should be noted that client authentication to the ISP is not related to authentication carried out by the PPTP server.

  The PPTP server controls all access to the private network by requiring a standard Windows NT-based logon. All PPTP clients must supply a Windows NT compatible user-id and password. As the PPTP tunnel is encrypted this is as secure as logging on from a computer connected physically to the private LAN.

  Authentication of PPTP clients is done using PPP authentication mechanisms. This includes the *Microsoft Challenge Handshake Authentication Protocol* (MS-CHAP), and the *Password Authentication Protocol* (PAP) authentication schemes.

- *Access Control* – Once the remote client has been authentication, all access to the private LAN is determined through access control mechanisms available under the Windows NT security

model. For example, access to resources on NTFS drives, or to other network resources requires the remote client to have the proper permissions.

- *Data Encryption* – PPTP uses a "shared-secret" cryptographic protocol. Both the PPTP client and server share a secret-key, which is derived from the user's password. The RSA RC4 standard is used to create a 40-bit session-key based on the user's hashed password. This key is then used to encrypt all the data contained in a PPP packet. The PPP packet containing a block of encrypted data is then encoded with a slightly modified version of GRE and encapsulated into a larger IP datagram for routing over the Internet to the PPTP server. Figure 7-10 shows the encapsulation of each protocol packet involved in a PPTP tunnel. Interception of the IP datagram would only reveal media headers (e.g. GRE), IP headers, and the PPP packet containing a block of encrypted data.

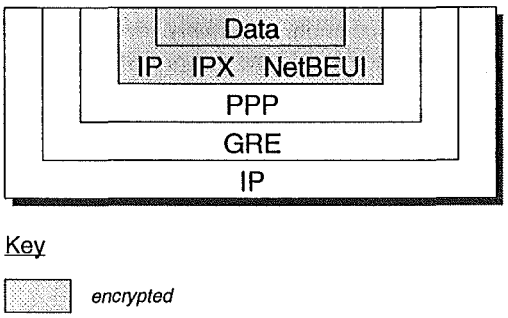

Figure 7-10 IP datagram containing encapsulated PPTP packets.

Due to cryptographic export controls only PPTP servers and clients supporting 40-bit session-keys are available outside of the US. However, users in the US and Canada can obtain a cryptographic pack which increases the session-key length to 128-bits.

7.6 IP Security (IPSec)

IPSec provides security at the IP-level and addresses the problems of authentication, integrity, and confidentiality. The authentication mechanism ensures that the IP datagram was actually sent by the party identified from the source IP address contained in the datagram header. The same mechanism also ensures the integrity of the datagram, i.e. it has not been altered in transit. The confidentiality mechanism ensures through encryption that a datagram's content is meaningless to any party except the sender and receiver(s).

In August 1995, the IETF IP Security Working Group published five proposed standards which define a set of requirements for IP-level security. Together this set of standards is known as IP Security, or IPSec. The IPSec documents are:

- RFC 1825 — Security Architecture for the Internet Protocol
- RFC 1826 — IP Authentication Header
- RFC 1827 — IP Encapsulating Security Payload
- RFC 1828 — IP Authentication using Keyed MD5
- RFC 1829 — The ESP DES-CBC Transform

Support for the features described in these standards is mandatory for IPv6 and optional for IPv4. In either case the security features are implemented as extension headers that follow the main IP header. The extension header for authentication is referred to as the *authentication header* (AH); while the header for confidentiality is referred to as the *encapsulating security payload* (ESP). Both headers are described below.

7.6.1 Security Association

Authentication and confidentiality rely on *security associations* which establish the context for the communication, and may define the security aspects of that communication. An association is a one-way relationship between a sender and receiver. Two security associations are required if a peer relationship is needed to establish secure two-way communication.

A security association is uniquely identified by an Internet destination address and a *security parameter index* (SPI). Therefore, the security association is uniquely identified by the destination address in the IPv4 or IPv6 header and the SPI contained within the AH or ESP header.

A security association normally includes the parameters listed below, but might include additional parameters as well [Atkinson, 1995a]:

- Authentication algorithm and algorithm mode being used with the IP AH (required for AH implementations)
- Key(s) used with the authentication algorithm in use with the AH (required for AH implementations)
- Encryption algorithm, algorithm mode, and transform being used with the IP ESP (required for ESP implementations)
- Key(s) used with the encryption algorithm in use with the ESP (required for ESP implementations)
- Presence/absence and size of a cryptographic synchronisation or initialisation vector field for the encryption algorithm (required for ESP implementations).
- Authentication algorithm and mode used with the ESP transform, if any is in use (recommended for ESP implementations).
- Authentication key(s) used with the authentication algorithm that is part of the ESP transform (if any) (recommended for ESP implementations).
- Lifetime of the key or time when key change should occur (recommended for all implementations).
- Lifetime of this security association (recommended for all implementations).
- Source address(es) of the security association, might be a wildcard address if more than one sending system shares the same security association with the destination (recommended for all implementations).
- Sensitivity level (e.g. secret or unclassified) of the protected data (required for all systems claiming to provide multi-level security, recommended for all other systems).

It is important to note that the SPI is the only feature which relates the key management mechanism used to distribute keys to the authentication and confidentiality mechanisms. Hence, the mechanisms specified for authentication and confidentiality are independent of specific key management mechanisms.

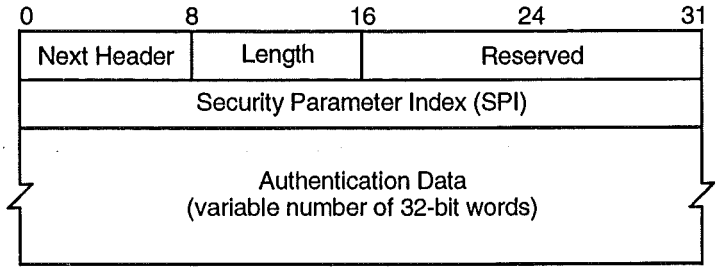

Figure 7-11 Authentication header.

7.6.2 Authentication

The AH provides the support for data integrity and authentication. The AH consists of the following fields [Atkinson, 1995b] (see Figure 7-11):

- *Next header* (8-bits) – identifies the next payload after the authentication payload.
- *Payload Length* (8-bits) – the length of the Authentication Data field in 32-bit words. Minimum value is 0 words, which is only used in the case of a "null" authentication algorithm.
- *Reserved* (16-bits) – reserved for future use. Must be set to all zeros when sent. The value is included in the authentication data calculation, but is otherwise ignored by the recipient.
- *Security Parameters Index* (32-bits) – a 32-bit pseudo-random value identifying the security association for this datagram. The SPI value 0 is reserved to indicate that "no security association exists".
- *Authentication Data* (variable bit length) – the length of this field is variable, but is always an integral number of 32-bit words. Many implementations require padding to other alignments, such as 64-bits, in order to improve performance.

The contents of the authentication data field depend on the authentication algorithm specified. The entire IP datagram, excluding fields that may change in transit, is used to calculate the authentication data. Fields that may change during transit are set to zero for the purpose of calculation at both source and destination.

For IPv6 the default authentication algorithm is MD5 (see Section 7.3.3), and all hosts must support this algorithm. If both parties agree, an alternative algorithm may be used. It should be noted that all hosts must support authentication, but they are not required to use it.

To compute the authentication data using MD5, the sender and receiver must share a secret authentication key (or *key*). If the key is shorter than 128-bits, it is padded out with zeroes (referred to as *keyfill*) to a length of 128-bits. The IP datagram is then appended to the 128-bits of key and keyfill. The appended IP datagram should also have an authentication header, however the authentication data is set to zero. All fields in the IP datagram that may change in transit are temporarily set to zero. Finally, the key is appended to the IP datagram.

This process results in a block of data constructed from the following sequence:

key + keyfill + IP datagram + key

The block of data is then passed to the MD5 algorithm, which generates a 128-bit result referred to as the message digest. The resulting message digest is used as the authentication data within the AH.

When the IP datagram is received the same steps are performed. The resulting message digest is then compared to the authentication data contained in the original AH. If both match then the sender is authentic and the IP datagram's integrity is intact. However, if there is a difference then either the sender is not who they claim to be, or the IP datagram was altered while in transit. No other party can generate an IP datagram that will be successfully authenticated unless they obtain a copy of the secret-key.

7.6.3 Confidentiality

Authentication is important for the security of IP datagrams, unfortunately it does not protect against the most basic security threat — eavesdropping. As IP datagrams traverse a network they may travel through many different systems and networks. Any of these intermediate nodes may harbour an attacker able to monitor the datagrams travelling through their domain. It is a simple task to attach a software or hardware protocol analyser to a network.

IPSec provides confidentiality through the *encapsulating security payload* (ESP). Depending on the senders requirements, this mechanism can be used to encrypt either a transport-layer segment (e.g. TCP, UDP, or ICMP), known as *transport-mode* ESP, or an entire IP datagram, known as *tunnel-mode* ESP.

An ESP header starts with a 32-bit SPI, with the remainder of the header (if any) containing parameters necessary to the encryption algorithm being used. The SPI and encryption parameters are generally transmitted as plaintext, while the remainder of the header is encrypted.

All IPSec systems that implement ESP must support the cipher block chaining (CBC) mode of DES as the default encryption algorithm. In CBC mode, the plaintext is processed as a sequence of 64-bit blocks. The input of the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block — the same key is used throughout. This has the effect of chaining together each block of plaintext. The benefit of chaining is that the ciphertext block bears no resemblance to the original plaintext block, therefore repeated blocks of plaintext are not exposed. Producing the first block of ciphertext requires an initialisation vector (IV) to be XORed with the first block of plaintext.

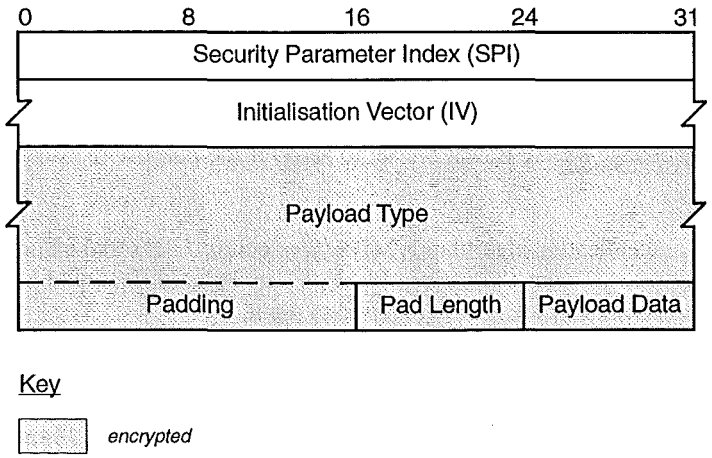

Figure 7-12 Encapsulating Security Payload (ESP) format.

Figure 7-12 shows the format of the ESP header including the encrypted payload data using DES-CBC. The fields are as follows:

- *Security Parameter Index* (32-bits) – identifies a security association
- *Initialisation Vector* (variable bit length) – initial input required for DES-CBC whose length is a multiple of 32-bits
- *Payload Data* (variable bit length) – prior to encryption, contains the block of data to be encrypted, which may be a transport-layer segment (transport mode) or an IP datagram (tunnel mode)
- *Padding* (variable bit length) – prior to encryption, filled with unspecified data to align the pad length and payload type fields on a 64-bit boundary
- *Pad length* (8-bits) – the size of the unencrypted padding field
- *Payload type* (8-bits) – indicates the protocol type contained in the payload data field (e.g. IP, TCP)

Note that the IV is transmitted in plaintext. This is not the most secure approach, however, it is considered acceptable for the security provided by IPSec.

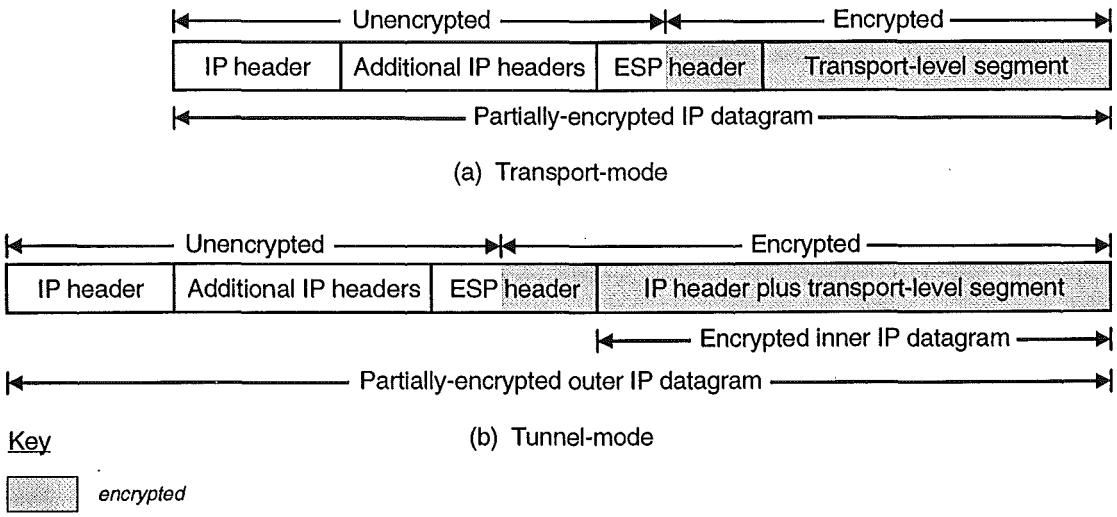

Figure 7-13 Secure IPv4 and IPv6 datagram.

Transport-Mode ESP

Transport-mode ESP encrypts the data carried by an IP datagram. This data is typically TCP or UDP which contains application-level data. In transport-mode, the ESP header is inserted into the IP datagram immediately prior to the transport-layer protocol header (e.g. TCP, UDP, or ICMP). In this mode bandwidth is conserved because there are no encrypted IP headers or IP options [Atkinson, 1995c].

Transport-mode operation is summarised as follows:

- The sender takes the original transport-layer (e.g. UDP, TCP, ICMP) frame and encapsulates it into the ESP. The sending user-id and destination address is used to locate the appropriate security association which determines the key and encryption algorithm to be used. If host-

oriented keying is in use, then all sending user-ids on a given system will have the same security association for a given destination address. If no key has been established, then the key management mechanism is used to establish an encryption key for the communication session prior to the encryption. The encrypted ESP is then encapsulated as the last payload of a plaintext IP datagram (see Figure 7-13a). The IP datagram can now be sent.

- The datagram is then routed to the destination. Intermediate routers simply route the datagram based on the IP header and optional plaintext IP headers. There is no need for the intermediate routers to examine the ESP.

- The receiver processes the plaintext IP header and optional plaintext IP headers. It then decrypts the ESP using the session key that has been established for this traffic, using the combination of the destination address and the datagram's SPI to locate the correct key. If no key exists for this session or the attempt to decrypt fails, the encrypted ESP is discarded and the failure recorded in the system or audit log. If decryption succeeds, the original transport-layer (e.g. UDP, TCP, ICMP) frame is removed from the ESP. The information from the plaintext IP header and transport-layer header is used to determine which application the data should be sent to.

Transport-mode provides an efficient and effective means of achieving confidentiality for any application that makes use of the IP protocol — individual applications need not implement their own confidentiality mechanisms. Unfortunately, transport-mode is vulnerable to traffic analysis as the destination address is never encrypted.

Tunnel-Mode ESP

Tunnel-mode ESP differs from transport-mode ESP in that the entire IP datagram is encrypted. The original IP datagram is placed in the encrypted portion of the ESP and that entire ESP frame is placed within a IP datagram having plaintext IP headers — needed so that the encrypted IP datagram can be routed to its destination. This method helps to limit traffic analysis because the final destination node cannot be determined by an eavesdropper.

Tunnel-mode ESP is particularly suitable for firewalls and other security gateways which protect trusted networks from untrusted networks. In this scenario, encryption is carried out between an external host and a security gateway, or between two security gateways. A major advantage of tunnel-mode is that key management is simplified by moving it from the internal hosts to the security gateway.

Tunnel-mode operation is summarised as follows:

- The sender takes the original IP datagram and encapsulates it into the ESP. The sending user-id and destination address are used to locate the correct security association to determine the appropriate encryption algorithm (and any associated parameters) to apply. If host-oriented keying is being used, then all sending user-ids on a given system will have the same security association for a given destination address. If no key has been established, then the key management mechanism is used to establish an encryption key for the session prior to use of ESP. The ESP is then encapsulated as the last payload of a plaintext IP datagram and sent (see Figure 7-13b).

- The IP datagram is routed to the destination by intermediate routers which make routing decisions based on the contents of the plaintext header. There is no need for the intermediate routers to examine the ESP — the ESP would be meaningless unless an attacker had compromised the session key.

- The receiver discards the plaintext IP header and any optional plaintext IP payloads. The combination of destination address and SPI value is used to locate the correct session key for decrypting the ESP. If no valid security association exists for this session (e.g. the receiver has no key), the receiver must discard the encrypted ESP and the failure must be recorded in the system or audit log. If decryption succeeds, the original IP datagram can be obtained from the

decrypted ESP. This original IP datagram is then processed as per the normal IP protocol specification, and may continue to be routed behind the security gateway.

7.6.4 Key Management

Each time an IP datagram is sent, the AH and ESP portion require a separate key. This means that 4 keys are required for a send and receive as both the AH and ESP require a pair of keys. For small sites key management can be achieved on a manual basis. However, in large sites where keys are being changed on a regular basis an automated key management system is highly desirable. With manual key management, keys for remote users and sites are produced and distributed by a central *Key Distribution Centre* (KDC). The key pair for transmit and receive are manually entered into the cryptographic systems at each end. Keys managed and distributed in this manner tend to only be changed on a weekly or monthly basis depending on the security of the site and the sensitivity of the communications.

Obviously, such a manual key management system is not suitable in circumstances where many users require keys which change constantly (e.g. every session, every datagram). Two key management protocols which can be implemented in IPSec for automated key production and distribution have been proposed:

- *Simple Key Management for Internet Protocols* (SKIP) – is a connection-less key management protocol. Prior communication is not necessary to carry out key management functions. To implement SKIP, each IP-based source and destination has a certified Diffie-Hellman (see Section 7.3.2) public-key. This public-key can be certified in various ways, including the use of X.509 (see Section 7.3.4) or PGP[53] certificates.

  Since all participants have access to each others public-key certificates, each participant has (implicitly) a mutually authenticated long-term secret-key with every other participant. This shared secret-key is not used to encrypt the IP datagram, instead a randomly generated packet-key is used — the shared secret-key becomes a *key-encrypting-key* (KEK) used to encrypt the packet-key (see Figure 7-14).

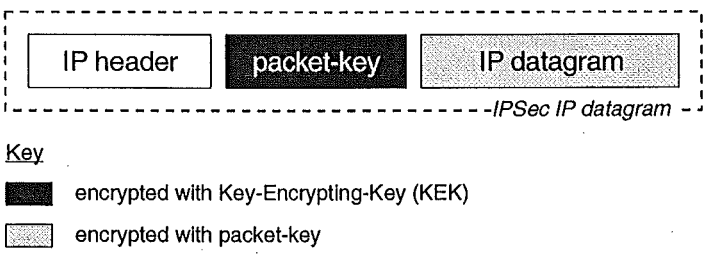

Figure 7-14 Encrypted IP datagram using SKIP.

  Since the KEK can be cached for efficiency, it allows the packet-keys to be modified very rapidly (e.g. per datagram) without incurring the computational overhead of a public-key operation [Aziz et al., 1995]. The strongest argument against SKIP is that it negates forward secrecy, i.e. if the keys are compromised then all previous traffic is exploitable.

- *Internet Security Association & Key Management Protocol* (ISAKMP) – is a common framework developed by the IETF IPSec working group to establish *Security Associations* (SA) and cryptographic keys in an Internet environment. SAs contain all the information required for execution of various network security services, such as the IP layer services (e.g. header authentication (AH) and payload encapsulation (ESP)), Transport or Application-layer

[53] PGP, an acronym for "Pretty Good Privacy", is an email security program designed by Philip Zimmermann in 1991. Information about PGP can be found at **http://www.pgp.com**

services, or self-protection of negotiation traffic. ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is distinct from key exchange protocols in order to separate the details of security association management (and key management) from the details of key exchange. There may be many different key exchange protocols (e.g. Diffie-Hellman, RSA etc.), each having different security properties. ISAKMP provides the common framework required for agreeing to the format of SA attributes, and for negotiating, modifying, and deleting SAs. ISAKMP also define basic requirements for its authentication and key exchange components which protect against denial-of-service, replay/reflection, man-in-the-middle, and connection hijacking attacks.

ISAKMP utilises digital signatures, based on public-key cryptography, for authentication. There are other strong authentication systems available, which could be specified as additional optional authentication mechanisms for ISAKMP. Some of these authentication systems rely on *trusted third-party* (TTP) KDCs to distribute secret session keys. An example is Kerberos, where the TTP is the Kerberos server which holds secret-keys for all clients and servers within its network domain. A client's proof that it holds its secret-key provides authentication to a server.

A generic key exchange protocol, known as Oakley, has been defined by the IPSec working group for use with ISAKMP. Oakley has several options for distributing keys. In addition to the classic DH exchange, this protocol can be used to derive a new key from an existing key and to distribute an externally derived key by encrypting it. The protocol can provide perfect forward secrecy, and permits the use of authentication based on symmetric encryption or non-encryption algorithms. This flexibility is provided to enable parties to choose the features that are most suited to their security and performance requirements.

Standardisation is required for all automated key distribution systems because the underlying network equipment must be able to interact with a centralised key management system at some point. Although the IETF chose ISAKMP with Oakley, Sun Microsystems which developed SKIP is working with a number of vendors (including Internet Dynamics, Novell, OpenROUTE, Swiss Federal Institute of Technology, Elvis+, CheckPoint, Toshiba, VPNet, Information Resource Engineering, Fortress Networks) to position SKIP as the *de facto* standard.

7.7 Secure Sockets Layer

The primary goal of the SSL protocol is to provide privacy and reliability between two communicating applications. The protocol is composed of two layers (see Figure 7-15). At the lowest level, layered on top of some reliable transport protocol (e.g. TCP), is the *SSL Record Protocol.* The SSL Record Protocol is used for encapsulation of various higher level protocols. One such encapsulated protocol, the *SSL Handshake Protocol,* allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data.

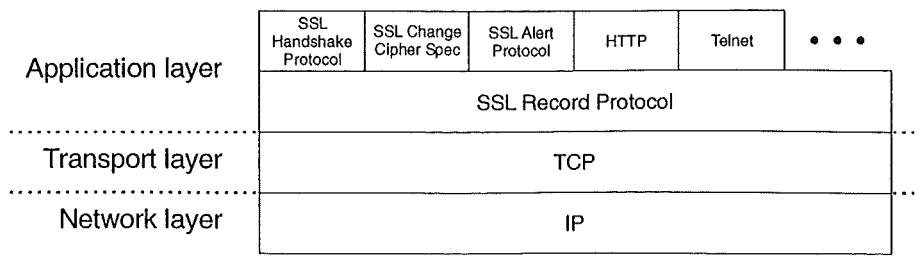

Figure 7-15 SSL protocol stack.

A major advantage of SSL is that it is application protocol independent. A higher level protocol can layer on top of the SSL Protocol transparently. The SSL protocol provides connection security that has three basic properties:

- The connection is private. Public-key cryptography is used after the initial handshake to define a secret-key. The secret-key is then used with the negotiated symmetric algorithm (e.g. DES, RC4, etc.) to encrypt the Application-layer data.

- The peer's identity can be authenticated using asymmetric, or public-key, cryptography (e.g. RSA, DSS, etc.)

- The connection is reliable. Message transport includes a message integrity check using a keyed *Message Authentication Code* (MAC). Secure hash functions (e.g. SHA, MD5, etc.) are used for MAC computations.

SSL has been used primarily for encrypting sensitive information, such as credit card details, between WWW-browsers and Internet servers, and allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

Theoretically SSL can transparently secure any TCP based protocol running on any port if both parties know that each other is using SSL. Table 7-1 shows a number of well known services that have been implemented with SSL support, and have been assigned official port numbers (useful so firewalls, proxies, and gateways, know what type of traffic to expect).

Table 7-1 Services implemented with SSL support.

Service	Port Number	Description
https	443	hyper-text transfer protocol
ssmtp	465	SMTP mail
snews	563	NNTP news
ssl-ldap	636	LDAP directory
spop3	995	POP3 mail
ftps	990	FTP — file transfer

SSL has undergone several revisions (see Table 7-2) since its initial release as SSL version 2.0 (SSL 2.0) in 1995. SSL version 3.0 (SSL 3.0) was the last official release supported by Netscape, and was submitted in 1996 to the IETF's Transport Layer Security (TLS) working-group as an Internet-Draft. Subsequently, 1997 saw the emergence of the TLS protocol standard [Dierks et al., 1997], version 1.0 (TLS 1.0), from the TLS working-group. Essentially TLS 1.0 is a modified definition of SSL 3.0, and although the modifications are significant enough to prevent interoperation it does incorporate a mechanism by which a TLS implementation can back down to SSL 3.0).

Table 7-2 History of SSL and derived protocols.

Version	Source	Description	WWW-browser Support
SSL 2.0	Published by Netscape.	Formed the original protocol specification.	Netscape 3.0, 4.0 Internet Explorer 3.0, 4.0
SSL 3.0	Published by Netscape, and expired IETF Internet-Draft.	Revised to prevent specific security attacks, add ciphers, and support certificate chaining.	Netscape 3.0, 4.0 Internet Explorer 3.0, 4.0
TLS 1.0	Expired IETF Internet-Draft	Based on SSL 3.0 with a number of modifications. SSL3.0 and TLS 1.0 are not interoperable.	None

7.7.1 Session Establishment

The SSL session is established by following a handshake sequence between the SSL client and the SSL server, as shown in Figure 7-16. This sequence may vary, depending on whether the server is configured to provide its public-key certificate (certificates used for SSL are usually of X.509v3 format, see Section 7.3.4) or request the client's public-key certificate. When the client and the server first start communicating, they must agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public-key cryptography to generate shared secrets. These processes are performed as part of the handshake protocol, which is summarised as follows:

- The client sends a *ClientHello* message to which the server must respond with a *ServerHello* message, or else a fatal error will occur and the connection will fail. The ClientHello and ServerHello are used to establish the security capabilities of the client and server. The ClientHello and ServerHello establish the following attributes: *protocol version*, *session ID*, *cipher suite* (e.g. symmetric and asymmetric algorithms, keys, cryptographic parameters, etc.), and *compression method*.

- Following the hello messages, the server sends its public-key certificate, if it is to be authenticated. Additionally, a *ServerKeyExchange* message may be sent, if it is required (e.g. if their server has no certificate, or if its certificate is for signing only). If the server is authenticated, it may request the client's public-key certificate if that is appropriate to the cipher suite selected. Now the server will send the *ServerHelloDone* message, indicating that the hello-message phase of the handshake is complete. The server will then wait for a client response.

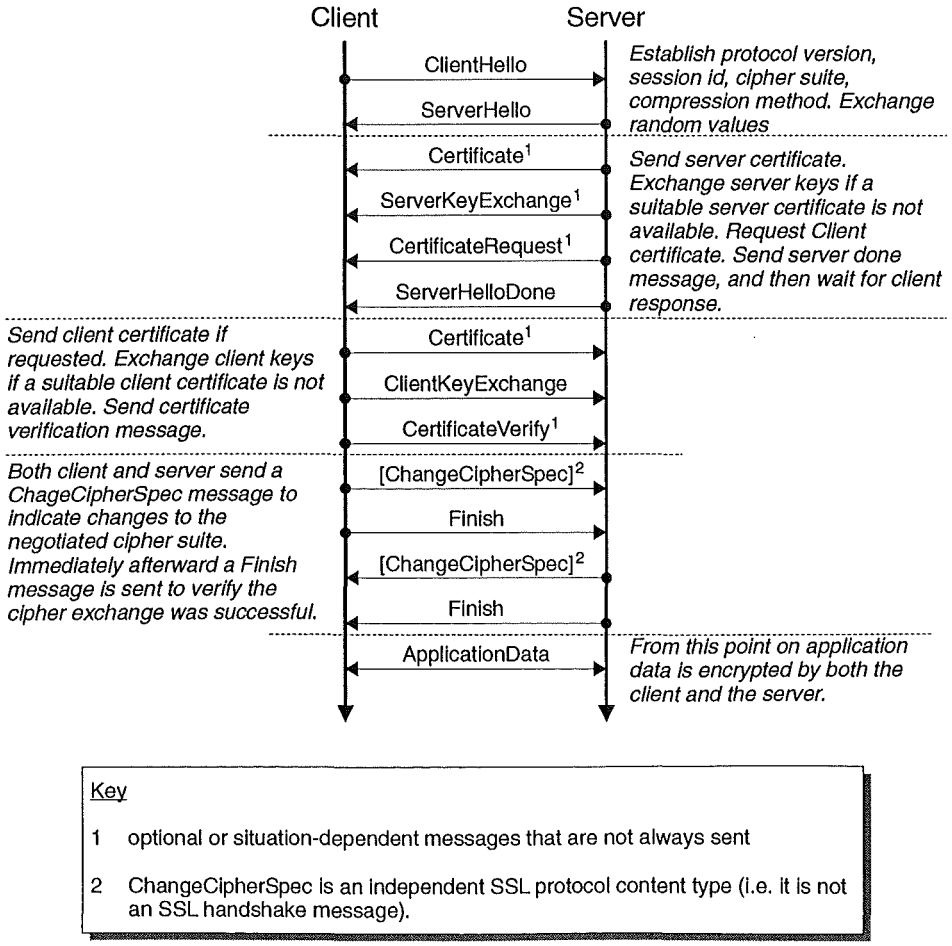

Figure 7-16 SSL handshake sequence.

- If the server has sent a *CertificateRequest* message, the client must send either the certificate message (containing the client's public-key) or a *no_certificate* alert. The ClientKeyExchange message is now sent, and the content of that message will depend on the public-key algorithm negotiated with the ClientHello and the ServerHello. If the client has sent a certificate with signing ability, a digitally-signed *CertificateVerifyMessage* is sent to explicitly verify the certificate.

- At this point, a *ChangeCipherSpec* (used to change the parameters of the current cryptographic protocols) message is sent by the client, and the client copies the pending *CipherSpec* (contains the parameters for the negotiated cryptographic protocols) into the current CipherSpec. The client then immediately sends the *Finish* message using the new algorithms, keys, and secrets. In response, the server will send its own ChangeCipherSpec message, transfer the pending to the current CipherSpec and send its Finish message under the new CipherSpec.

- At this point, the SSL handshake is complete — the client and server may now begin exchanging Application-layer data in a secure manner.

7.7.2 Data Transfer

The SSL Record protocol (See Figure 7-17) is used to transfer application and SSL control data between the client and the server, possibly fragmenting this data into smaller units, or combining multiple higher layer protocol messages into single units. It may also compress, attach message digests, and encrypt units before transmitting them using the underlying reliable transport protocol. Once the SSL handshake is complete, the two parties have shared secrets which are used to encrypt compressed record units and compute MACs on their contents.

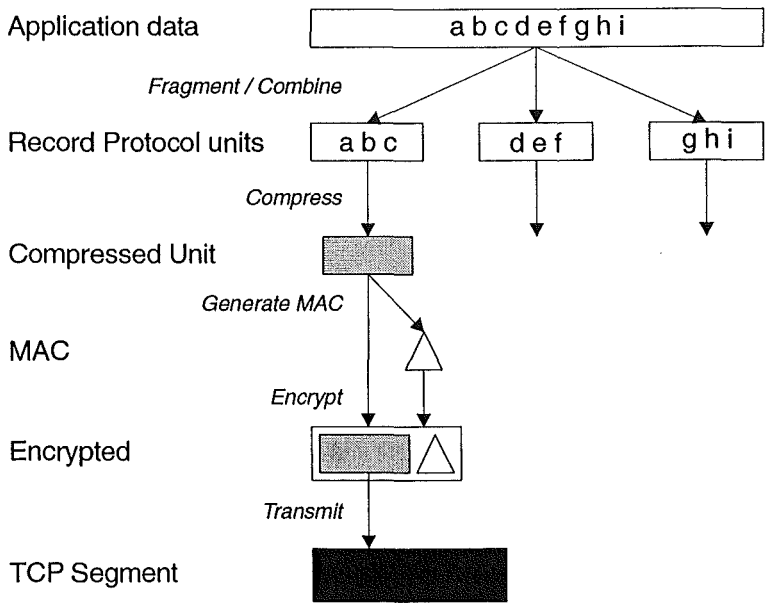

Figure 7-17 SSL record protocol.

Data from higher layers (see Figure 7-15) is fragmented into *SSLPlaintext* records of 2^{14} bytes or less, and delivered to the SSL Record protocol. All SSLPlaintext records are compressed using the compression algorithm defined in the current session state. There is always an active compression algorithm, however initially it is defined as *CompressionMethod.null* (i.e. initial handshake records are not compressed). The compression algorithm translates an SSLPlaintext unit into an SSLCompressed unit. Compression must be lossless and may not increase the content length by more than 1024 bytes. If

the decompression function encounters a fragment that would decompress to a length in excess of 2^{14} bytes it will issue a fatal alert message.

All records are protected using the encryption and MAC algorithms defined in the current CipherSpec. There is always an active CipherSpec, however initially it is SSL_NULL_WITH_NULL_NULL, which does not provide any security. The techniques used to perform the encryption and MAC operations are defined by the CipherSpec. The encryption and MAC functions translate an SSLCompressed structure into an SSLCiphertext. The decryption functions reverse the process. Most implementations of SSL, by default, implement the following nine symmetric algorithms for encrypting compressed record units:

- No encryption
- Stream Ciphers
  - ◊ RC4 with 40-bit keys
  - ◊ RC4 with 128-bit keys
- CBC Block Ciphers
  - ◊ RC2 with 40-bit keys
  - ◊ DES40, DES, 3DES_EDE
  - ◊ Idea
  - ◊ Fortezza

The choice of hash function determines how a message digest is created from a record unit. SSL supports the following:

- No digest
- MD5
- SHA-1

The message digest is used to create the MAC which is encrypted with the message to provide integrity and to prevent against replay attacks. Transmissions also include a sequence number so that missing, altered, or extra messages are detectable.

7.7.3 SSL and Proxies

One problem with SSL-based connections is that they do not work with proxy servers. For a proxy server to support SSL it must either support SOCKS[54] (which stands for "sockets"), or use a special SSL Tunnelling protocol. Both of these options are supported in Netscape's Proxy Server.

SSL was designed to provide security between client and server and to avoid any kind of 3-way man-in-the-middle attack. Thus SSL cannot be proxied through traditional application-level firewalls (such as the CERN proxy server), because SSL considers a proxy server to be a middleman.

The simplest solution for this problem is to use a packet filtering firewall configured to allow a reserved and trusted port to be opened for each SSL enabled service, such as HTTPS or SNEWS (443 or 563 respectively). Effectively this allows all traffic on those ports to be passed through unrestricted. The risk however, with this solution, is that an internal attacker could attempt to use these trusted ports without using SSL and there is no way for the firewall to know.

[54] The SOCKS protocol is an independent proxy mechanism that provides a generic byte forwarding gateway between a client and a server, and generally works at the socket level. SOCKS is sufficient if the only requirements are for TCP/UDP restrictions based on client or server IP addresses. Information about SOCKS can be found at **http://www.socks.nec.com**

However, most non-SSL HTTP proxies work at the protocol level and have the ability to understand header information related to the protocol. This goes beyond SOCKS to allow the firewall administrator to use the header information to filter and/or monitor the traffic. SOCKS simply does not provide enough information about a request to let a firewall decide whether to allow it or to log the request.

A more secure approach is to use a firewall that supports the SSL Tunnelling CONNECT extension method as described in [Luotonen, 1995].

In SSL Tunnelling, the client initiates an SSL connection via normal HTTP, then handshakes and creates a secure connection to the server via a byte-forwarding tunnel. The proxy has access to the client-proxy request headers, but the session is encrypted. Once the handshake occurs, the proxy acts just like a SOCKS gateway. This allows the firewall to monitor the requests, but not the traffic.

There are three additional things that the SSL Tunnelling mechanism does with the proxy server that do not happen when using SOCKS:

- The client sends a "user agent" message (for example, "Mozilla/3.0/Macintosh").
- The proxy can send to the client an authorisation request allowing the administrator to use passwords to control external Internet access.
- The standard is more easily extensible. For example, the client could, in theory, send the URL being requested (or anything else) to the firewall. However, there is no standard to support this behaviour.

Another solution (also available using the Netscape Proxy Server), is that the proxy server can spoof[55] SSL on behalf of the internal client. The proxy will initiate SSL between itself and other servers on the Internet, but be unsecure inside the firewall between the proxy server and the client.

This compromise means that client authentication is not possible; only server authentication of the remote sites is available. However, it does provide the ability for client authentication between the client and the proxy. The administrator must decide which is more important, until such time as a better solution arises.

It is also possible for a proxy server to hold both client and server keys for its internal clients. This allows SSL sessions to be carried out twice: once between the client and proxy server, and again between the proxy server and the secure server. Thus, the proxy server can listen in on the conversation without having the private-keys of external servers. Clearly this is not reasonable for the general Internet, but it is a viable solution for corporate requirements such as Intranets and Extranets.

7.8 Summary

VPNs are an essential tool for protecting information as it travels through both trusted and untrusted networks. With the increasing numbers of organisations using the Internet as an extension of their private networks, VPN technologies provide a way for organisations to protect their information. In addition to providing confidentiality, VPNs can also provide non-repudiation, integrity, and strong authentication services. Different combinations of these services are required for Intranets, Extranets, and Teleworkers.

The basis for VPNs is cryptography, which forms the mathematical basis for each of the aforementioned services. Secret-key cryptography has been used for many years, unlike public-key cryptography which has a relatively short history. Regardless of its youth public-key cryptography has revolutionised many areas of computer security, including connectionless (e.g. email) and connection-

[55] Information about Netscape Proxy Server's ability to spoof SSL on behalf of an internal client can be found at **http://developer.netscape.com/library/one/sdk/proxy/unixguide/ssl-tunl.htm#518342**

oriented (e.g. Telnet) communications. However, this Chapter has focused on connection-oriented VPN technologies, such as PPTP, IPSec, and SSL.

Currently each of these particular VPN technologies fills a particular role in connection-oriented communication. PPTP allows dial-in access to the organisations private networks, and allows the user's computer to appear as if it were a part of the internal network. SSL is most often used to create dynamic VPNs on a per-session basis (e.g. between a WWW-browser and a WWW-server), and has found a niche protecting WWW-based credit-card transactions in the continuing absence of a finalised SET standard and implementation. Although, it is expected that this particular SET will eventually replace SSL because it provides specialised benefits such as automating the entire credit-card transaction process. IPSec will probably become the omnipresent VPN technology, unlike SSL and PPTP which operate at the Transport and Application-layer, IPSec provides cryptographic and key-management mechanism to the Network-Layer. The benefit of IPSec is that it provides long overdue security mechanisms to the Internet, and its related protocols.

Chapter 8. Conclusions

8.1 The future of Internet Security

The security of the Internet has long been in question, and is now under greater scrutiny than ever before. There is currently a great deal of debate, especially in the US, as to what the threats are and where they are coming from. The US relies heavily on the Internet and is concerned that the infrastructure that supports it is becoming an attractive target for its enemies — this thinking has coined the term "Information Warfare", in which the objective is to disrupt (or destroy) an enemies information systems. The US is taking the possibility of Information Warfare very seriously, and has convened numerous government committees to investigate the impact and likelihood of such attacks. The following story from the Washington Times [Brosnan, 1998] paints a very interesting picture of the possible impact that Information Warfare could have:

> *"A band of seven hackers from Boston told a Senate Committee yesterday that they could bring down the foundations of the Internet in 30 minutes. Testifying under their Internet aliases -- Mudge, Brian Oblivion, Space Rogue, Kingpin, Weld Pond, John Tan and Stefan Von Neumann -- the hackers said that by interfering with the links between long-distance phone carriers such as AT&T and MCI they could disrupt Internet service for a couple of days.*
>
> *The hackers, known collectively as LOpht, opened a series of hearings by Senate Governmental Affairs Committee Chairman Fred Thompson, Tennessee Republican, on the security of government and commercial computer and telecommunication networks. Mr. Thompson released a pair of reports by the congressional General Accounting Office that said the State Department and the Federal Aviation Administration's air control system are highly vulnerable to hacking.*
>
> *In a test, congressional investigators accessed the travel itineraries of U.S. diplomats, employment records and e-mail traffic and were even able to take control of the State Department's computers. Much of the FAA report was so scary it was classified. Utilities, stock exchanges, the Federal Reserve and taxpayer credit and medical records also are at risk, Mr. Thompson said. 'It seems the more technologically advanced we've become the more vulnerable we've become,' he said. "Our nation's underlying information infrastructure is riddled with security flaws."*
>
> *The LOpht hackers blamed the poor security on the patchwork nature of the Internet networks, government laxity and the indifference of makers of operating systems and software to security concerns. 'Simple security measures are missing from almost all the software sold to companies today,' Mudge said. For instance, while Microsoft claims its Windows NT server for businesses is more secure than Windows 95 for personal users, Weld Pond said hackers usually can break into an NT system in less than a day. Mr. Thompson predicted it is only a matter of time before Microsoft and other software makers find themselves being sued by a company whose system has been penetrated through their software.*
>
> *Not all the testimony was bleak. The hackers said it is far easier to interfere with service than to change data or issue commands. For instance, the Global Positioning Satellite system used in military and some civilian aircraft for navigation can be jammed, but it is unlikely a hacker could move a satellite's position, the hackers testified. Still, Space Rogue said, a determined group of hackers could 'wreck havoc in the country.'..."*

Although this thesis has not looked specifically at the issue of Information Warfare, the threats and vulnerabilities discussed in Chapter 3 would be applicable for use in such a "war". It has also been shown, through numerous examples and statistical data from widely conducted surveys, that organisations connected to the Internet face threats very similar to those described above. However,

threats do not simply come from attackers external to the organisation, rather a great deal of computer crime is conducted from inside the organisation by "trusted" employees. As it is often easier to launch attacks from within an organisation, there is a very real possibility of Information Warfare being waged from within. To help protect against such attacks and the threat of Information Warfare increasing numbers of organisations are installing firewall architectures and VPNs.

Firewall architectures have evolved to provide a means of segregating networks that have incompatible security policies or represent different levels trustworthiness. The main benefit of a firewall architecture is that it reduces the zone-of-risk the organisation's internal networks are exposed to, and it provides a central point for the implementation and management of security policy.

Although firewall architectures provide very good protection from attacks launched against the Network and Transport-layers, the unpredictability and complexity found at the Application-level significantly reduce the type of protection that can be achieved by a firewall. In fact, firewalls are vulnerable to incorrectly implemented applications, and provide little (if any) protection against threats from viruses and malicious executable content (e.g. Java applets, ActiveX controls).

The greatest problem with firewall technology is that it provides no protection for traffic once it is sent onto a network. It also has no way of authenticating traffic that it receives, unless it relies on some mechanism at the Application-layer. To provide solutions to these problems VPN technology is being integrated with firewall architectures.

VPNs provide a number of security mechanisms such as confidentiality, message authentication, non-repudiation, and message integrity. Although these mechanisms could be provided solely through symmetric or asymmetric cryptography, for efficiency and saleability reasons hybrid (combines the best features of both symmetric and asymmetric algorithms) cryptographic systems are most often used. With the increasing use of Intranets, Extranets, and teleworkers there is a growing need for VPN technologies to be implemented.

In general, it should be expected that firewall architectures and VPNs will continue to merge until their functionality is inseparable. The use of VPNs can reduce and even eliminate many classes of attack — sniffing a network for passwords will be impossible, as will IP spoofing and other forms of address impersonation — however, many new problems will undoubtedly arise. For example, public-key certificate servers will need to be very secure and will be tempting targets for internal abuse. In addition, denial-of-service attacks may even become more predominant.

This thesis has also shown the benefit of having firewall architectures verified and validated by independent parties — a process known as certification. Without such certification a purchaser has only vendor assurance that the firewall architecture will function correctly, and is not vulnerable to compromise — the assumption made for firewalls is that they cannot be compromised! Chapter 6 has reviewed the current and future direction of firewall certification schemes driven by both government and commercial organisations. It is widely accepted that current firewall technology is not advanced enough to protect systems that process highly sensitive (e.g. nationally classified) information. This is reflected by the fact that no firewall has been evaluated above the ITSEC E3 assurance level. Commercial certification has focused on penetration testing, using the same tools that are generally available to attackers from the Internet. Obviously, such evaluation can be automated but it does not provide the same assurance as evaluations carried out under government certification schemes that use well developed criteria and take a much more comprehensive look at the firewall architecture.

8.2 Problems and Future Research

This thesis has identified many areas that pose problems for firewall and VPN technologies. The following points discuss some of these problems and suggest areas of future research:

- *Key Management* – for both SSL (and the future TLS protocol) and IPSec a common *public-key infrastructure* (PKI), or certificate hierarchy, is required to provide ubiquitous interoperable support for authentication and encryption. The problem is that no such global

infrastructure exists at present. In WWW-browsers that support SSL this problem has been addressed somewhat by distributing with the browser a number of public-key certificates from well known organisations (e.g. Microsoft, Netscape, etc.) In addition, SSL also supports anonymous DH key exchanges which provide confidentiality but not authentication.

IPSec can also support anonymous DH key exchange, and certificates could be sent manually to all of the organisations that wish to carry out authentication of the connections they accept. However, scaling this approach to the global Internet would not be practicable. The aim of IPSec is to provide global support for confidentiality and authentication at the IP layer, which can be achieved between hosts that until the connection was initiated, had no prior knowledge of each other. The most obvious place to implement a global PKI would be in the existing DNS. Public-key certificates could then be exchanged at the same time as the DNS resolver query. Unfortunately, DNS is an untrusted service and it still remains to be seen who would (regardless of the DNS being used) sign all of the certificates. Obviously there are many areas here that will require further research before IPSec will be widely deployed, and of any great use.

- *End-to-end Security* – this is the most obvious area in which security mechanisms, such as non-repudiation, confidentiality, integrity etc., are required to protect information transmitted over unsecure connections. A great deal of work has already been done to accomplish such mechanisms within VPN technologies (e.g. SSL, PPTP, and IPSec). However, more research needs to be done in policy negotiation between network domains, and the related problem of proxying encrypted traffic through firewalls (or gateways).

- *End-System Security* – the majority of security problems observed within the Internet are related directly to insufficient or incorrect security implementations at end points (i.e. hosts). Many examples of such problems have been presented through this thesis, including, sendmail, Ping of Death, Java applets, etc. In addition to the many security incidents caused by software bugs, poor user administration can provide easy targets for attackers (e.g. allowing trust relationships through .rhosts files). Unfortunately, these problems are not addressed sufficiently, if at all, by firewall architectures and VPNs. In fact, these problems stem mainly from the lack of correct software engineering methodologies being applied to system development. A great deal of work remains to be done to improve the standard of software development at all layers of the TCP/IP suite (in particular the Application-layer).

- *Secure QoS* – the pending addition of QoS features to the Internet as part of IPv6 introduces a new set of security issues. In particular IPv6 provides flow label and priority fields [Bradner, 1995] that allow hosts to specify special handling of traffic by IPv6 routers, for example the host may be participating in a "real-time" connection that requires defined latency and bandwidth parameters. This capability is particularly important in order for IPv6 to support multimedia, and other applications that require some degree of consistent throughput or delay. Obviously, firewalls would provide a convenient location for controlling and protecting the provision of services so that users cannot utilise more resources than they are authorised to, or deny services to users who have legitimate requirements. The users of these services must be authenticated to ensure that the services are being consumed by the users for which they are intended.

- *Secure Network Infrastructure* – The network itself must be protected — this is one of the major targets of Information Warfare. The validity of the routing and control messages must be assured in order for the Internet to function reliably. The exchange of routing information between routers must be authenticated in order to prevent false information from being inserted into the routing tables and disrupting traffic — in effect denial-of-service attacks!

- *Internal Attacks* – Insider abuse accounts for a considerable amount of computer crime, and firewalls architectures are unable to deal effectively with such attacks. Part of the problem lies with security policy, the NSAP and FAP need to be designed with a focus on both internal and external security — often the focus is only on protecting external abuse. In particular, services should only be allowed through a firewall provided they are absolutely necessary for an

employee to carry out their daily tasks. In addition, the combination of VPN, certificate based authentication mechanisms, and sufficient auditing, implemented on the internal network would prevent a great deal of the abuse. It may also be necessary to integrate intrusion detection systems as part of the firewall architecture.

- *Encryption Policy* – Encryption policy and its interpretation present problems in a number of areas. For example, the adoption of mandatory cryptographic protocols within IPSec has implications for export control and usage functionality. The export of encryption technology is restricted by a number of governments, including the New Zealand government. Some governments (e.g. Russia) take cryptographic control even further by prohibiting private citizens using any encryption products. Such policies make it difficult to promote a standard set of cryptographic protocols, and for manufacturers to export their products. In addition, situations could arise where encrypted traffic blocked from a network domain because of government controls on cryptography.

The Internet will evolve to incorporate integrated firewall and VPN architectures that will replace traditional dedicated WAN links. This is possible because VPN technology provides vital security services, including, confidentiality, integrity, non-repudiation, and strong authentication.

Appendix A

ITSEC Target Evaluation Levels

The requirements for each assurance level are outlined below and detailed in Chapter 4 of the ITSEC [ITSEC, 1991]. The requirements at each level build on those of the previous level.

Assurance Level	Description
E0	represents inadequate assurance and cannot be claimed for any TOE undergoing evaluation, as the TOE may only achieve E0 as a result of an evaluation.
E1	requires a security target and an informal description of the architectural design of the TOE. Functional testing should be performed to show that the TOE satisfies the security target. User and administration documentation must give guidelines on maintaining product security. In addition, the TOE must be uniquely identified and have delivery, configuration, start-up and operational documentation. There must be evidence that secure distribution methods have been utilised.
E2	requires an informal description of the detailed design and test documentation. The separation of security enforcing and other components must be shown within the architectural design. The developer's configuration control procedures, and security measures adopted to maintain the integrity of the product, will be assessed. Audit trail output, if produced during start-up and operation, is required.
E3	requires source code or hardware drawings in relation to SEFs and security relevant functions, with an informal description of the correspondence from these to the detailed design. There must be evidence that acceptance procedures are used and that retesting has occurred after the correction of errors. The implementation languages used should conform to recognised standards.
E4	requires a formal model of the TOE's security policy along with a semi-formal specification of the SEFs, architectural and detailed design. There must be evidence that testing covers all SEFs in sufficient detail. The TOE, and any tools used, must be under configuration control with any changes being audited. All compiler options should be documented. The TOE must retain its security after restart as a result of failure.
E5	requires the architectural design to explain the interrelationship between security enforcing components, with close correspondence between the detailed design and source code/hardware drawings. There must be information on the integration process and run time libraries. Configuration control must be independent of the developer. Configured items must be identified as security enforcing or security relevant, with support for variable relationships between them.
E6	requires a formal description of the architecture and SEFs. There must be correspondence from the formal specification of the SEFs through to source code and tests. Different TOE configurations shall be defined in terms of the formal architectural design. All tools shall be subject to configuration control.

Appendix B

NCSA FWPD Criteria

Version 2.0

I. Functionality Requirements

The product under test will be will be installed and configured to support the service requirements listed below. In the event that multiple means are available for supporting a feature, the "most transparent" mode will be used for supporting internal users, and the "most restrictive" used in supporting external users.

Operating system facilities which do not directly support the service/security requirements of the firewall will be disabled insofar as is possible.

In instances where multiple means of supporting a certification-required function are available within a firewall product's capabilities, the vendor may provide recommendations as to NCSA as to its preferred configuration for meeting such requirements.

Information regarding the configuration of products at the time of certification will be considered non-proprietary.

1. Services to Internal Clients

- Telnet through firewall to external networks (1)
- FTP through firewall to external networks (1)
- HTTP through firewall to external networks (2)
- *SSL **and/or** SHTTP through the firewall to external networks.*
- SMTP mail through firewall to external networks
- DNS - external DNS information must be made available to internal clients

Legend:

1. *SOCKS may be used in meeting these requirements.*
2. *HTTP may be provided via a proxy/cache or by filter.*

2. Services provided to External Users

- FTP access to a server located on the internal network *or a service network.*(1) (2) (3)
- HTTP access to a server located on the internal network *or a service network.*(3)
- *SSL **and/or** SHTTP access to a server located on the internal network.*
- SMTP mail must be deliverable to clients on internal networks
- DNS - some form of "presence" must be configurable

Legend:

1. *FTP service need not be anonymous*, **and may require authentication**
2. *If an authentication key device is required to access a service, the vendor must supply the device to NCSA for testing.*

3. *"Internal" FTP or HTTP/SSL/SHTTP servers may be located on a service network if the product supports it. A service network is a additional network attached to the firewall which exists for the purpose of supporting such servers.*

3. Firewall Management

The console of the firewall system must be securable, requiring a password authentication for access.

If a remote management capability is provided for use over external networks:

- a one-time password mechanism **or other secure means of authentication must be utilized**
- an encrypted link mechanism must be utilized

If a remote management capability is provided for use over internal networks IP address must not be the sole mechanism for administrator authentication.

II. Security Requirements

Upon demonstration of the functional requirements in section I, the configuration under test will be subjected to the following tests. The tests will be tuned utilizing full knowledge of the test configuration and its components; reducing the impact of "security by obscurity". *The same set of attacks will be mounted from the internal network, as if an attacker had gained access to a system on the internal network.*

To receive certification, a product (as configured to meet the requirements of section I. above) must resist all attacks listed in this section, in accordance with the following criteria:

- No measure of administrative control of the firewall or the underlying operating system may become available to the attacker as a result of the attacks applied.
- No protocol or data content other than that specified in section I. must pass the firewall and be carried on the internal network.
- Denials of service: The product under test must not be **trivially** rendered inoperable by network-based attacks with the following exceptions:
- The product has a documented fail-safe mechanism for removing itself from service according to a declared policy.
- **If a denial of service attack is widely recognized as having no defense, the product must provide a log-based alert prior to failing.**

Products which do not meet these criteria will not be certified. [The FWPD Certification Contract is the authoritative document governing administrative procedures for certification, resolution of certification problems, certification usage, and decertification.]

During testing, network monitors will be utilized both on the protected network and on the segment outside of the firewall.

1. ISS Security Scanner

The most current production version (and/or interim release) of the ISS Security Scanner product will be configured with full knowledge of the firewall and systems it is protecting. All possible

modes/attacks will be configured and enabled regardless of applicability to the configuration under test.

The ISS Security Scanner represents an aggregate of common threats known and repeatable at the time of its release. NCSA will continually update to the latest production release and/or interim release of this product as part of its ongoing testing.

External Scan

The scanner will be run against the configuration under test from a non-adjacent subnet.

Internal Scan

The scanner will be run against the configuration from a "trusted" internal system.

2. Port Scanning

A port scanning tool will be run against the configuration under test, for the purpose of determining conformance to the service requirements in section I. Scans will be run from both the "trusted" internal net and an untrusted non-adjacent subnet.

Information from the scans will be compared with the service requirements (Section I).

3. NCSA Tools

As part of its network security advocacy role, NCSA collects and builds tools for use in penetration testing and vulnerability assessment. Generally these tools incorporate emerging attack methodologies, and or demonstration code for publicised exploits.

NCSA will apply tools from its inventory to the configuration under test based on OS-type, firewall type, and active ports/services (as revealed by II.2).

In the event of a successful exploitation which compromises the firewall platform or the protected network, the vendor will be provided with information regarding the attack's "signature" and on-wire protocol traces. In only the rarest of cases (where such methods do not provide the vendor with sufficient information to remedy the problem) will NCSA consider releasing the tool which facilitated the breach of security. Any such release of code would require execution of an NCSA Malicious Code Agreement*.

Appendix C

Privileged Port Numbers

Services which can be provided by any implementation of TCP/IP are designated port numbers between 1 and 1023, these are known as *privileged ports*. The privileged ports are managed by the *Internet Assigned Numbers Authority* (IANA). Before 1992 privileged ports were restricted to between 1 and 255, with UNIX specific services taking ports 256 to 1023 [Stevens, 1994].

The following is a list of privileged and unprivileged TCP/IP ports. Unprivileged ports are used by non-standard or application specific services and take numbers greater than 1023.

Table 8-3 Privileged and unprivileged port numbers.

Port Number	Protocol	Service Name	Comment	Firewall Action
1	TCP	*tcpmux*	TCP port multiplexor	Block.
7	UDP, TCP	*echo*	Echoes UDP datagrams and characters sent down TCP streams.	Block, can be used for denial-of-service attacks, and probing of the network.
9	UDP, TCP	*discard*	Accepts connections, but discards the data.	Block.
11	TCP	*sestet*	Returns active users on the system. Can be connected to *Whois*. Used to gather information on likely targets.	Block
13	UDP, TCP	*daytime*	Returns human-readable time of day. Used to compromise security protocols based on the systems real-time clock.	Block.
15	TCP	*netstat*	Officially unassigned.	Block.
17	UDP	*qotd*	Returns quote of the day.	Block.
19	UDP, TCP	*chargen*	Generates a continuous cycling, character stream.	Block.
20	TCP	*ftp (data)*	Data port for FTP. Vulnerable to sniffing attacks.	Allow — requires special consideration.
21	TCP	*ftp (control)*		
23	TCP	*telnet*	Telnet — character based remote terminal. Vulnerable to sniffing attacks.	Allow — requires consideration.
24	UDP, TCP		Used by private email systems.	Block.
25	TCP	*smtp*	Email	Allow — requires consideration.
37	UDP, TCP	*time*	Returns machine-readable time of day. Can be used to	Block.

Port Number	Protocol	Service Name	Comment	Firewall Action
			compromise security protocols based on the systems real-time clock.	
38	UDP, TCP	*rap*	Route Access Protocol.	Block.
42	UDP, TCP	*name*	Host Name Server — Obsolete.	Block.
43	TCP	*whois*	Usually run by Network Information Centres.	Allow In — if running a sanitised Whois server. Allow Out, or Block.
48	UDP, TCP	*auditd*	Digital Equipment Corporation audit daemon	Block.
49	UDP	*tacacs*	Used to authenticate logins to terminal servers. Vulnerable to sniffing and spoofing attacks.	Block, should be reachable only from the internal side of the firewall.
53	UDP, TCP	*domain*	Domain Name Service	Use separate name servers for internal and external use. If proxies are used then DNS service only required on firewall.
67	UDP, TCP	*bootps*	Bootstrap protocol server.	Block.
68	UDP, TCP	*bootpc*	Bootstrap protocol client.	Block.
69	UDP	*tftp*	Trivial FTP.	Block.
70	TCP	*gopher, gopher+*	Text based information service. Vulnerable to sniffing attacks.	Allow In — only to secured organisational Gopher server. Allow Out — with proxies. Block — if not demanded.
79	TCP	*finger*	Returns information on user account or host machine. Used by attacker to gather information on user accounts and hosts.	Allow In — to sanitised message. Allow Out. Block — if not demanded.
80	TCP	*http*	Used for World Wide Web access. Vulnerable to sniffing and spoofing attacks.	Allow Out — with proxies. Allow In — to organisational WWW-server running on special host.
87	TCP	*link*		Block.

Port Number	Protocol	Service Name	Comment	Firewall Action
88	UDP	*kerberos*	A distributed authentication mechanism.	Block, unless inter-domain authentication is required.
94	UDP, TCP	*objcall*	Tivoli Object Dispatcher	Block.
95	TCP	*supdup*	Virtual terminal similar to Telnet. Vulnerable to sniffing attacks.	Block.
109	TCP	*pop-2*	Post Office Protocol used to remotely transfer mail. Vulnerable to sniffing attacks.	Block unless demanded. Allow in — to mail host only.
110	TCP	*pop-3*	The latest version of Post Office Protocol.	
111	UDP, TCP	*sunrpc*	Sun Remote Procedure Call portmapper. Vulnerable to spoofing attacks.	Block.
113	TCP	*auth*	Authentication service which identifies the username of a TCP connection. Vulnerable to spoofing attacks.	Block, or limit to use between trusted domains.
119	TCP	*nntp*	Network News Transport Protocol.	Block, or limit to use between trusted hosts.
121	UDP, TCP	*erpc*	Encore Expedited Remote Procedure Call.	Block.
123	UDP, TCP	*ntp*	Network Time Protocol. Vulnerable to spoofing attack.	Block, or limit to use between trusted hosts.
126	UDP, TCP	*unitary*	Unisys Unitary Login.	Block.
127	UDP, TCP	*locus-con*	Locus PC-Interface Conn Server.	Block.
130	UDP, TCP	*cisco-fna*	Cisco FNATIV	Block, unless required by Cisco hardware.
131	UDP, TCP	*cisco-tna*	Cisco TNATIVE	
132	UDP, TCP	*cisco-sys*	Cisco SYSMAINT	
137	UDP, TCP	*netbios-ns*	NetBIOS Name Service	Block, or limit to use between trusted domains. Use an encrypted NetBIOS tunnel over TCP/IP.
138	UDP, TCP	*netbios-dgm*	NetBIOS Datagram Service.	
139	UDP, TCP	*netbios-ssn*	NetBIOS Session Service.	
144	UDP, TCP	*news*	Sun NeWS (Network Window System) is an obsolete service. Vulnerable to sniffing and spoofing attacks.	Block.

Port Number	Protocol	Service Name	Comment	Firewall Action
156	UDP, TCP	*sqlsrv*	SQL Service. Vulnerable to sniffing attacks.	Block.
161	UDP, TCP	*snmp*	Simple Network Management Protocol. Vulnerable to spoofing and sniffing attacks.	Block.
162	UDP, TCP	*snmptrap*	SNMP traps.	Block. Allow from gateway to internal network monitors.
177	UDP, TCP	*xdmcp*	X Display Manager (XDM) Control Protocol. Vulnerable to spoofing and sniffing attacks.	Block, unless demanded or in special conditions.
178	UDP, TCP	*NSWS*	NEXTSTEP Window Server. Vulnerable spoofing and sniffing attacks.	Block.
194	UDP, TCP	*irc*	Internet Relay Chat Protocol.	Block, unless demanded.
199	UDP, TCP	*SMUX*	SMUX (IBM).	Block.
200	UDP, TCP	*src*	IBM System Resource Controller.	Block.
201	UDP, TCP	*at-rtmp*	AppleTalk Routing Maintenance.	Block, or limit to use between trusted domains. Use an encrypted AppleTalk tunnel over TCP/IP.
202	UDP, TCP	*at-nhp*	AppleTalk Name Binding.	
203	UDP, TCP	*at-3*	AppleTalk Unused.	
204	UDP, TCP	*at-echo*	AppleTalk Echo.	
205	UDP, TCP	*at-5*	AppleTalk Unused.	
206	UDP, TCP	*at-zis*	AppleTalk Zone Information.	
207	UDP, TCP	*at-7*	AppleTalk Unused.	
208	UDP, TCP	*at-8*	AppleTalk Unused.	
210	TCP	*wais*	WAIS Server. Vulnerable to sniffing attacks.	Block, unless a server is being run.
220	TCP	*imap*	POP replacement. Vulnerable to sniffing attacks.	Block.
387	TCP	*avrp*	AppleTalk Routing.	Block.
396	UDP, TCP	*netware-ip*	Novell Netware over IP. Vulnerable to sniffing attacks.	Block.
411	UDP, TCP	*rmt*	Remote Tape.	Block.

Port Number	Protocol	Service Name	Comment	Firewall Action
512	UDP	*biff*	Real-time mail notification.	Block.
512	TCP	*exec*	Remote command execution. Vulnerable to sniffing attacks.	Block.
513	UDP	*rwho*	Remote Who command.	Block.
513	TCP	*login*	Remote Login. Vulnerable to spoofing and sniffing attacks.	Block, unless demanded. Replace with secure login
514	UDP	*shell*	Remote Shell (rsh). Vulnerable to spoofing and sniffing attacks.	mechanisms. Vulnerable to problems with "trusted hosts" and .rhost files.
514	TCP	*syslog*	Used for passing *syslog* messages.	Block. Allow, if necessary, from gateway to internal security monitors.
515	TCP	*printer*	Berkeley *lpr* system. Vulnerable to spoofing attacks.	Block.
517	UDP	*talk*	Initiate *talk* requests.	Block, unless
518	UDP	*ntalk*	Initiate *talk* requests.	demanded or for special circumstances. Difficult to control connections.
520	UDP	*route*	Used to control routing. Vulnerable to spoofing attacks.	Block.
523	UDP, TCP	*timed*	Time server daemon. Vulnerable to spoofing.	Block.
532	UDP, TCP	*netnews*	Remote *readnews*.	Block.
533	UDP, TCP	*netwall*	Network Write to all users.	Block.
540	TCP	*uucp*	Commonly used to transfer Usenet news. Vulnerable to spoofing and sniffing attacks.	Block, or limit to use between trusted hosts.
550	UDP, TCP	*nrwho*	New *rwho*.	Block.
566	UDP, TCP	*remotefs*	Remote File System. Vulnerable to spoofing and sniffing attacks.	Block.
666	TCP	*mdqs*	Replacement for Berkley's printer system.	Block.
666	UDP, TCP	*doom*	Network Doom — game.	Block.
744	TCP	*FLEXlm*	FLEX license manager.	Block.
754	TCP	*tell*	Used by send.	Block.

Port Number	Protocol	Service Name	Comment	Firewall Action
755	UDP	*securid*	Security Dynamics ACE/Server. Vulnerable to sniffing.	Block, unless demanded. Encryption can be disabled by administrator. Non US versions do not provide encryption.
765	TCP	*webster*	Dictionary service.	Block.
1025	TCP	*listener*	System V Release 3 listener.	Block.
1352	UDP, TCP	*lotusnotes*	Lotus Notes mail system.	Block.
1525	UDP	*archie*	Used to search the Internet for resources.	Block, unless a server is being run.
2000	TCP	*OpenWindows*	Sun proprietary window system.	Block.
2049	UDP, TCP	*nfs*	Sun Network File System (NFS). Vulnerable to spoofing.	Block.
2766	TCP	*listen*	System V listener.	Block.
3264	UDP, TCP	*ccmail*	Lotus cc:Mail.	Block.
5130	UDP	*sgi-dogfight*	Silicon Graphics flight simulator.	Block.
5133	UDP	*sgi-bznet*	Silicon Graphics tank demo.	Block.
5500 5510	UDP TCP	*securid* *securidprop*	Security Dynamics ACE/Server version 2. Vulnerable to sniffing attacks. Security Dynamics ACE/Server slave. Vulnerable to sniffing attacks.	Block, unless demanded. Encryption can be disabled by administrator. Non US versions do not provide encryption.
5701	TCP	*xtrek*	X11 xtrek.	Block.
6000 ↓ 6063	TCP	*X-server*	X11 server. Vulnerable to spoofing and sniffing attacks	Block
6667	TCP	*irc*	Internet Relay Chat.	Block, unless demanded.
7000 ↓ 7009	UDP, TCP	*afs*	Andrew File System (AFS).	Block.
7100	TCP	*font-service*	X Server font service.	Block.

References

[Amoroso, 1994] Amoroso, E. 1994. Fundamentals of Computer Security Technology. Prentice-Hall.

[Atkinson, 1995a] Atkinson, R. 1995. *Security Architecture for the Internet Protocol.* RFC 1825, August.

[Atkinson, 1995b] Atkinson, R. 1995. *IP Authentication Header.* RFC 1826, August.

[Atkinson, 1995c] Atkinson, R. 1995. *IP Encapsulating Security Payload (ESP).* RFC 1827, August.

[Aziz et al., 1995] Aziz, A; and Patterson, M. 1995. *Simple Key-Management for Internet Protocols (SKIP).* Proceedings of the 5th Annual Conference of the Internet Society, Hawaii, June 27–30.

[Barkow, 1996] Barkow, T. 1996. *The Domain Name System — let your resolver do the walking.* Wired, September, pg. 84.

[Bellovin, 1989] Bellovin, S. 1989. *Security Problems in the TCP/IP Protocol Suite.* Computer Communication Review, Vol. 19, No. 2, April, pp. 32–48.
Available at **ftp://ftp.research.att.com/dist/internet_security/ipext.ps.z**

[Borman, 1993] Borman, D. 1993. *Telnet Authentication Option.* RFC 1416, February.

[Bradner, 1995] Bradner, S (Ed.). 1995. *IPng — Internet Protocol Next Generation.* Addison-Wesley Publishing Company, pp. 233–237.

[Brosnan, 1998] Brosnan, J. 1998. *Hackers testify they can crash Internet service in a half-hour.* Washington Times, May 20.

[Brownlee, 1994] Brownlee, N. 1994. *New Zealand Experiences with Network Traffic Charging.* ConneXions, Vol. 8, No. 12, December.
Available at **http://www.auckland.ac.nz/net/Accounting/nze.html**

[Bryant et al., 1997] Bryant, D; and Brittain, P. 1997. *DLSw v2.0 Enhancements.* RFC 2166, June.

[Caceres et al., 1991] Caceres, R; Danzig, P; Jamin, S; and Mitzel, D. 1991. *Characteristics of Wide-Area TCP/IP Conversations.* Computer Communication Review, Vol. 21, No. 4, September, pp. 101–112.

[Cafarchio, 1998] Cafarchio, P. 1998. Personal correspondence, March 3. Mr. Cafarchio is currently employed by the International Computer Security Association (ICSA) as Firewalls Program Manager.

[Carl-Mitchell et al., 1993] Carl-Mitchell, S; and Quarterman, J. 1993. *Practical Internetworking with TCP/IP and UNIX.* Addison-Wesley Publishing Company.

[CCITSE, 1996] 1996. *Common Criteria for Information Technology Security Evaluation.* Common Criteria Editorial Board, version 1.0 (Draft), January 31.
Available at **http://www.tno.nl/instit/fel/refs/cc.html**

[CCITT, 1988] 1988. *Data Communication Networks: Service and Facilities, Interfaces.* Proceedings of the CCITT IXth Plenary Assembly Melbourne, Vol. VIII, Fascicle VIII.2, November 14–25, pp. 251–256.

[CEM, 1997] 1997. *Common Evaluation Methodology for Information Technology Security*. Common Criteria Editorial Board, Version 0.6 (Draft), November 1.
Available at **http://www.tno.nl/instit/fel/refs/cc1.0/cem1_971.ps**

[CFS, 1996a] 1996. *Computer Fraud and Security*. Elsevier Science Ltd., February, pg. 3.

[CFS, 1996b] 1996. *Computer Fraud and Security*. Elsevier Science Ltd., April, pg. 4.

[Cheswick et al., 1994] Cheswick, W; and Bellovin, S. 1994. *Firewalls and Internet Security – Repelling the Wily Hacker*. Addison-Wesley Publishing Company.

[Cisco, 1997] 1997. *Solutions for Virtual Private Dialup Networks*. Cisco Systems Inc., January 2.
Available at **http://www.cisco.com/warp/public/728/General/vpdn_wp.htm**

[Cohen, 1998] Cohen, A. 1998. Personal correspondence, February 2 to March 28. Mr. Cohen founded JOTA System Security Consultants Inc. of Ontario, Canada. He is currently a Canadian ISO representative working on the Common Criteria.

[Computer Weekly, 1995] 1995. Computer Weekly, November 16.

[Computerworld, 1998] 1998. *Top Kiwi Companies Say Yes to Teleworking*. Computerworld, No. 535, February 9, pg. 1.

[Costales et al., 1993] Costales, B; Allman, E; and Rickert, N. 1993. *Sendmail*. O'Reilly and Associates Inc.

[CPL, 1997] 1997. *Certified Products List*. Certification Body of the UK IT Security Evaluation and Certification Scheme, October.
Available at **http://www.itsec.gov.uk/docs/pdfs/products.pdf**

[Cray, 1997] Cray, A. 1997. *Secure VPNs Lock the Data, Unlock the Savings*. Data Communications, May 21, pp. 49–56.

[CSI, 1998] 1998. *1998 CSI/FBI Computer Crime and Security Survey*. Computer Security Institute.
Available at **http://www.gocsi.com/prelea11.htm**

[DeMaio, 1995] DeMaio, H. 1995. *Internet Security: Connecting Without Fear*. Info Security News (Supplement), Published by Michael I. Sobol.

[Dierks et al., 1997] Dierks, T; and Allen, C. 1997. *The TLS Protocol Version 1.0*. Internet Engineering Task Force, Internet-Draft, November 12.

[Ellison et al., 1997] Ellison, C; Frantz, B; Lampson, B; Rivest, R; Thomas, B; and Ylonen, T. 1997. *Simple Public Key Certificate*. Internet Engineering Task Force, Internet-Draft, November 21.

[EM1, 1997] 1997. *Description of the AISEP*. Australian Information Security Evaluation Programme, Evaluation Memorandum No. 1, Issue 1.1, March.

[EM7, 1997] 1997. *Developers Guide*. Australian Information Security Evaluation Programme, Evaluation Memorandum No. 7, Issue 1.0, June, pg. 11.

[Garfinkel, 1996] Garfinkel, S; and Spafford, G. 1996. *Practical UNIX and Internet Security*. O'Reilly & Associates Inc., 2nd Edition, April.

[Guha et al., 1997] Guha, B; and Mukherjee, B. 1997. *Network Security via Reverse Engineering of TCP Code: Vulnerability Analysis and Proposed Solutions*. IEEE Network, July/August, pp. 40–48.

[Hamzeh et al., 1997a] Hamzeh, K; Pall, G; Verthein, W; Taarud, J; and Little, W. 1997. *Point-to-Point Tunneling Protocol — PPTP*. Internet Engineering Task Force, Internet-Draft, July.

[Hamzeh et al., 1998] Hamzeh, K; Rubens, A; Kolar, T; Littlewood, M; Palter, B; Valencia, A; Taarud, J; Townsley, W; Pall, G; and Verthein, W. 1998. *Layer Two Tunnelling Protocol "L2TP"*. Internet Engineering Task Force, Internet-Draft, January.

[Hamzeh, 1997b] Hamzeh, K. 1997. *Ascend Tunnel Management Protocol — ATMP*. RFC 2107, February.

[Hanks et al., 1994a] Hanks, S; Li, T; Farinacci, D; and Traina, P. 1994. *Generic Routing Encapsulation (GRE)*. RFC 1701, October.

[Hanks et al., 1994b] Hanks, S; Li, T; Farinacci, D; and Traina, P. 1994. *Generic Routing Encapsulation over IPv4 networks*. RFC 1702, October.

[Hare et al., 1996] Hare, C; and Siyan, K. 1996. *Internet Firewalls and Network Security*. New Riders Publishing, 2nd Edition.

[Harrenstien, 1977] Harrenstien, K. 1977. *Name/Finger Protocol*. RFC 742, December 30.

[Harrenstien, 1982] Harrenstien, K; and White, V. 1982. *Nicname/Whois*. RFC 812, March 1.

[Housley, 1993] Housley, R. 1993. *Security Label Framework for the Internet*. RFC 1457, May.

[Howes, 1995] Howes, T. 1995. *The Lightweight Directory Access Protocol: X.500 Lite*. University of Michigan, Center for Information Technology Integration (CITI) Technical Report 95–8, July 27.

[ITSEC, 1991] 1991. *Information Technology Security Evaluation Criteria, Provisional Harmonised Criteria*. Commission of the European Communities, Version 1.2, June 28.
Available at **http://www.itsec.gov.uk/docs/pdfs/ITSEC.PDF**

[Joncheray, 1995] Joncheray, L. 1995. *A Simple Active Attack Against TCP*. Proceedings of the Fifth USENIX UNIX Security Symposium, Salt Lake City, Utah, June 5–7, pp. 7–19.

[Kent, 1991] Kent, S. 1991. *Security Options for the Internet Protocol*. RFC 1108, November.

[Klensin, 1993] Klensin, F; Rose, T; Stefferud, E; and Crocker, D. 1993. *SMTP Service Extensions*. RFC 1425, February.

[Luotonen, 1995] Luotonen, A. 1995. *Tunneling SSL Through a WWW Proxy*. Internet Engineering Task Force, Internet-Draft, December 14.
Available at **http://www.netscape.com/newsref/std/tunneling_ssl.html**

[Menkus, 1995] Menkus, B. 1995. *Firewalls in Information Systems Security*. EDPACS, Vol. 23, No. 3, September.

[Microsoft, 1997] 1997. *Understanding PPTP*. Microsoft Corporation (White-Paper).
Available at **http://www.microsift.com/communications/exes/understand.exe**

[Mills, 1992] Mills, D. 1992. *Network Time Protocol (version 3) specification, implementation and analysis*. RFC 1305, March.

[Morris, 1985] Morris, R. 1985. *A Weakness in the 4.2BSD UNIX TCP/IP Software*. Computing Science Technical Report 117, AT&T Bell Laboratories, February 25.
Available at **ftp://netlib.att.com/netlib/research/cstr/117.z**

[NCSA, 1996] 1996. *National Computer Security Association (NCSA) Firewall Policy version 1.01.* Available from **http://www.ncsa.com**

[Neal, 1996] Neal, D. 1996. *The Harvest Object Cache in New Zealand.* WWW Journal, Issue 3. Available at **http://www.waikato.ac.nz/harvest/www5/Overview.html**

[NIST, 1995] 1995. *Secure Hash Standard.* National Institute of Standards and Technology, Federal Information Processing Standards Publication 180-1, April 17.

[Perkins, 1996a] Perkins, C. 1996. *IP Mobility Support.* RFC 2002, October.

[Perkins, 1996b] Perkins, C. 1996. *IP Encapsulation within IP.* RFC 2003, October.

[Phrack, 1996a] 1996. *Project Neptune.* Phrack Magazine, Vol. 7, Issue 48, July, File 13. Available at **http://www.phrack.com/Archives/phrack48.zip**

[Phrack, 1996b] 1996. *IP Spoofing Demystified: Trust Relationship Exploitation.* Phrack Magazine, Vol. 7, Issue 48, June, File 14. Available at **http://www.phrack.com/Archives/phrack48.zip**

[Plummer, 1982] Plummer, D. 1982. *An Ethernet Address Resolution Protocol.* RFC 826, November.

[Postel, 1980] Postel, J. 1980. *User Datagram Protocol.* RFC 768, August 28.

[Postel, 1981a] Postel, J. 1981. *Internet Control Message Protocol.* RFC 792, September.

[Postel, 1981b] Postel, J. 1981. *Internet Protocol.* RFC 791, September.

[Postel, 1981c] Postel, J. 1981. *Internet Control Message Protocol.* RFC 792, September.

[Postel, 1981d] Postel, J. 1981. *Transmission Control Protocol.* RFC 793, September.

[Postel, 1982] Postel, J. 1982. *Simple Mail Transfer Protocol.* RFC 821, August.

[Postel, 1983] Postel, J; and Reynolds, J. 1983. *Telnet Protocol Specification.* RFC 854, May.

[Postel, 1985] Postel, J; and Reynolds, J. 1985. *File Transport Protocol.* RFC 959, October.

[Ranum, 1993] Ranum, M. 1993. *Thinking About Firewalls.* Proceedings of the SANSII, Washington, DC. Available at **http://www0.tis.com/docs/products/gauntlet/ThinkingFirewalls.html**

[Rescorla et al., 1997] Rescorla, E; and Schiffman, A. 1997. *The Secure HyperText Transfer Protocol.* Internet Engineering Task Force, Internet-Draft, November.

[RSA, 1998] 1998. *RSA's Secret-Key Challenge Solved by Distributed Team in Record Time.* RSA Data Security (Press-Release), February 26. Available at **http://www.rsa.com/pressbox/html/980226.html**

[Safford et al., 1993a] Safford, D; Schales, D; and Hess, D. 1993. *The TAMU Security Package: An Ongoing Response to Internet Intruders in an Academic Environment.* Proceedings of the Fourth Usenix UNIX Security Symposium, Santa Clara, CA, October, pp. 91–118. Available at **http://www.tamu.edu/pub/mirrors/net.tamu.edu/tamu-security-overview.ps.gz**

[Safford et al., 1993b] Safford, D; Hess, D; and Schales, D. 1993. *Secure RPC authentication (SRA) for Telnet and FTP.* Proceedings of the Fourth Usenix UNIX Security Symposium, Santa Clara, CA, October, pp. 63–67.

[Scheifler et al., 1992] Sceifler, R; and Gettys, J. 1992. *X Window System.* Digital Press, 3rd Edition.

[Schneier, 1996] Schneier, B. 1996. Applied Cryptography. John Wiley & Sons Inc., 2nd Edition.

[Schultz, 1996] E. Eugene Schultz. 1996. *How to Perform Effective Firewall Testing.* Computer Security Journal, Vol. 12, No. 1, pp. 47–55.

[Showalter, 1996] Showalter, M. 1996. *Ascend Tunnel Management Protocol — ISP Applications.* Ascend Communications Inc. (White-Paper), May 21.
Available at **http://www.ascend.com/docs/techdocs/atmpisp.pdf**

[Sim et al, 1997] Sim, P; and Rudkin, S. 1997. *The Internet — past, present and future.* BT Technology Journal, Vol. 15, No. 2, April, pp. 11–23.

[Spafford, 1989] Spafford, E. 1989. *An analysis of the Internet worm.* Proceedings of the European Software Engineering Conference, September.
Available at **ftp://ftp.cs.purdue.edu/pub/spaf/security/IWorm.PS.Z**

[Stallings, 1991] Stallings, W. 1991. *Data and Computer Communications.* Macmillan Publishing Company, 3rd Edition.

[Stevens, 1994] Stevens, W. 1994. *TCP/IP Illustrated: the protocols.* Addison-Wesley Publishing Company.

[Stewart et al., 1997] Stewart, D; Maginnis, P; and Simpson, T. 1997. *Who is at the Door: The SYN Denial of Service.* Linux Journal, June, pp. 12–14, 16–17.

[Sun Microsystems, 1988] Sun Microsystems. 1988. *RPC: Remote procedure call protocol specification: Version 2.* RFC 1057, June.

[Sun Microsystems, 1989] Sun Microsystems. 1989. *NFS: Network File System Protocol Specification.* RFC 1094, March.

[TCSEC, 1985] 1985. *Department of Defense Trusted Computer System Evaluation Criteria.* National Computer Security Center, United States of America, Standard DOD 5200.28-STD, Library No. S225-711, December 26.
Available at **http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html**

[TPEP, 1998] 1998. *The Computer Security Evaluation Frequently Asked Questions (V2.1).*
Available at **http://www.radium.ncsc.mil/tpep/process/faq.html**

[UKSP04, 1996] 1996. *Roles of Developers in ITSEC.* UK IT Security Evaluation and Certification Scheme, UK Scheme Publication No. 4, Issue 1.0, July, pp. 11–39.

[UKSP06, 1997] 1997. *Certified Product List.* UK IT Security Evaluation and Certification Scheme, UK Scheme Publication No. 6, October.
Available at **http://www.itsec.gov.uk/docs/pdfs/products.pdf**

[Wells et al., 1995] Wells, L; and Bartky, A. 1995. *Data Link Switching.* RFC 1795, April.

[White et al., 1996] White, G; Fisch, E; and Pooch, U. 1996. *Computer System and Network Security.* CRC Press, pp. 9–22.

[Wiggin, 1996] Wiggin, P. 1996. *Wired Kiwis — Every New Zealander's Guide to the Internet.* Shoal Bay Press Ltd, pp. 18–25.
Available from **http://www.wiredkiwis.co.nz**

[X509, 1996] 1996. *ISO/IEC JTC1/SC 21, Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions.* December 1.

[Ylonen et al., 1997a] Ylonen, T; Kivinen, T; and Saarinen, M. 1997. *SSH Protocol Architecture.* Internet Engineering Task Force, Internet-Draft, November 7.

[Ylonen et al., 1997b] Ylonen, T; Kivinen, T; and Saarinen, M. 1997. *SSH Transport-layer Protocol.* Internet Engineering Task Force, Internet-Draft, November 7.

[Ylonen et al., 1997c] Ylonen, T; Kivinen, T; and Saarinen, M. 1997. *SSH Connection Protocol.* Internet Engineering Task Force, Internet-Draft, November 7.

[Ylonen et al., 1997d] Ylonen, T; Kivinen, T; and Saarinen, M. 1997. *SSH Authentication Protocol.* Internet Engineering Task Force, Internet-Draft, November 7.

[Zao et al., 1997] Zao, J; and Condell, M. 1997. *Use of IPSec in Mobile IP.* Internet Engineering Task Force, Internet-Draft, November.